Automatic Multilevel Feature Abstraction in Adaptable Machine Vision Systems

Thesis

Automatic Multilevel Feature Abstraction in Adaptable Machine Vision Systems

Valerie Rose BSc (Hons)

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

of the Open University

23rd December 2010

Department of Design, Development, Environment and Materials

Faculty of Mathematics, Computing and Technology

The Open University

Walton Hall

Milton Keynes

UK

Abstract

Vision is a complex task which can be accomplished with apparent ease by biological systems, but for which the design of artificial systems is difficult. Although machine vision systems can be successfully designed for a specific task, under certain conditions, they are likely to fail if circumstances change. This was the motivation for the research into ways in which systems can be self-designing and adaptable to new visual tasks. The research was conducted in three vital areas of concern for machine vision systems.

The first area is finding a suitable architecture for forming an appropriate representation for the current task. The research investigated the application of Hypernetworks theory to building a multilevel, generally-applicable representation, through repeated application of a fundamental 'self-similarity' principle, that parts of objects assembled under a particular relation at one level, form whole objects at the next. Results show that this is potentially a powerful approach for autonomously generating an adaptable system-architecture suitable for multiple visual tasks.

The second area is the autonomous extraction of suitable low-level features, which the research investigated through random generation of minimally-constrained pixel-configurations and algorithmic generation of homogeneous and heterogeneous polygons. The results suggest that, despite the simplicity of the features making them vulnerable to image transformations, these are promising approaches worth developing further.

The third area is automatic feature selection. The research explored management of 'dimensionality' and of 'combinatorial explosion', as well as how to locate relevant features at multiple representation levels, in the context of 'emergence' of structure. Results indicate that this approach can find useful 'intermediate-level' constructs through analysis of the connectivity of the simplices representing objects at higher levels.

The research concludes that the proposed novel approaches to tackling the above issues, in particular the application of hypernetworks to the formation of multilevel representations and the resulting emergence of higher-level structure, is fruitful.

Acknowledgements

Many people have contributed to the completion of this thesis and I thank them for their help, support and friendship.

I have been privileged to have as my supervisors, Jeff Johnson and Anthony Lucas-Smith, who have introduced me to research in the world of 'intelligent' machine vision systems. I thank them for their constant support and invaluable guidance throughout my journey of exploration into that world. As well as providing academic support, they have also helped ease travel problems for me by arranging to meet 'half way' on a regular basis for supervision meetings, in Glasgow. That has been greatly appreciated.

My thanks also go to Jane Bromley, Pejman Iravani and Joan Serras for their practical input and advice. In addition I thank Pejman and Joan for their encouragement to 'keep going' during the final writing-up stage.

I also thank my family and friends for their continual encouragement and patience.

I am especially grateful to my mother, whose support was unfailing, and who was always a willing listener despite the unfamiliar subject matter. Sadly, she did not live to see the completion of the work.

Finally, I thank my husband, John, who has been and continues to be my 'rock'. His support, both emotional and practical, has been vital in enabling me to complete the project. I dedicate this work to him.

Valerie Rose December 2010

Contents

List of Figures

List of Tables

Chapter 1. Introduction: motivation and research aims

The thesis is concerned with the design of machine vision systems that can learn to represent and recognize many different categories of object with minimal user input, and that can adapt to changes in visual task requirements without having to be redesigned 'from scratch'.

1.1 The problem of representation and recognition in automatic machine vision systems

Vision is a complex task. It is an ill-posed problem in that, for a given scene, there are many possible interpretations of its contents. In the human visual cortex, many different scenes can give rise to very similar neural responses. Conversely, the same object can elicit different patterns of neural response when it appears under varying lighting conditions, at a different scale, in an alternative location, at an altered orientation, or when closely surrounded by, or even partially obscured by, other objects. Yet, the human visual system is capable of resolving ambiguity in a continually changing visual environment, from which often only noisy or incomplete information is available. This is a rapid process, achieved apparently effortlessly, enabling us to arrive at an appropriate conclusion, much of the time, about what is 'out there', so that relevant action can be instigated, for example, evading a perceived threat, appreciating a work of art, or picking up the car keys from the kitchen table.

This perceptive ability has evolved over millions of years, and is not only dependent on a highly-structured, multi-layered system for representing and constructing the information coming from the eye, but also relies on its synthesis with stored knowledge or experience in memory.

Representation is a crucial part of the process of visual perception, and is the focus of the thesis in its exploration of how artificial visual systems can adapt themselves to new visual tasks. A *representation* is the result of converting an incoming 'signal' from the environment into sets of measurement 'values' that can readily be interpreted by the system as indicative of the presence of certain objects or 'features' within the scene. The process of obtaining these measurements is

often referred to as '*feature extraction*'. The measurements the system takes must be relevant to the task in hand. For example, it is doubtful that colour is a useful measurement for distinguishing a lemon from a banana, whereas length is likely to be more appropriate. Choosing features that are relevant for the given task is commonly termed '*feature selection*'. The *multilevel* nature of the representation that has evolved in biological vision is an important consideration. Hubel and Wiesel (1962) (Hubel 1995) hypothesize that primate vision is hierarchical with the representation becoming increasingly complex at successive levels through the combination of 'features' at lower levels. The thesis explores how machine vision systems can build and adapt multilevel representations for different visual tasks.

Machine vision systems clearly lack the advantage of long-term evolution, and have limited ability to learn. Nevertheless, they are employed successfully in a large number of domains bringing with them the benefits of reduced human work-load, with improved operational efficiency and reliability and often considerable financial savings. Their application ranges from medical imaging and diagnostics, through security applications such as face and fingerprint recognition, to finding images or documents relating to a particular topic on the internet.

Such systems are generally designed by a programmer so that the representational architecture and the method of pre-processing the data are optimal for a specific visual task and classification technique. The consequence of this is that, when faced with a task outside the intended application, many systems are unable to adapt, and thus perform poorly with the new data.

Hence the overall aim of the work of the thesis has been to increase the autonomy and adaptability of machine vision systems in object recognition.

For a machine vision system to be able to adapt itself to cope with many different visual tasks, it would need to find a way of designing its own architecture, and of finding features suited to the given task at each representational level. This realization led to the formulation of a set of

'important' requirements for an adaptable machine vision system. These are introduced in the next section.

1.2 Requirements of an adaptable system

Biological vision systems have evolved a multilevel architecture in response to the demands for rapid and reliable interpretation of the environment required for an animal's survival.

This suggests that a system that can adapt needs a general architectural framework in which it can autonomously extract information from images at multiple levels of complexity.

All that a system user should be required to do is to 'point' at the object or region of interest in a scene, say by outlining it, Figure 1.1, and then the system should be able to:

- Extract suitable low-level structural configurations
- Define spatial connectivities among parts
- Abstract higher-level constructs
- Integrate the different levels of representation

At the same time, the system must overcome three significant and related problems:

- Combinatorial explosion
- The curse of dimensionality
- The intermediate word problem

Research in biological vision, reviewed in Chapter 2 of the thesis, has shed light on how primate vision tackles these problems.

'Combinatorial explosion' occurs when there are too many possible choices of how to combine patterns at one level to form higher-level constructs at the next. This then leads to the 'curse of dimensionality' problem at the new level. Biological vision avoids combinatorial problems by requiring *local* connectivity for passing information between representation levels. This reduces the number of possible connections between levels and implicitly encodes spatial information

about how different parts of a scene or object are positioned relative to one another (Wallis and Rolls, 1997).

The 'curse of dimensionality' occurs when, at any representational level, there is an excessive number of patterns or descriptors, so that objects or object parts are being represented in a very high-dimensional space, which, especially if the number of training examples is relatively small, can result in a rather sparse, widely-spread distribution with no clear clustering of classes. This makes classification difficult. Often, many of the dimensions are irrelevant for the given task. It is thought that in biological vision, one way that irrelevant features are eliminated is that patterns of response 'compete' with one another at multiple levels to form an appropriate representation (Rolls and Deco, 2002).

The 'intermediate word' problem (Johnson, 2006) is that, given that the lowest and highest representations are known, how should lower-level constructs be combined and higher level constructs be disaggregated to form intermediate-level representations, and how many such levels should there be? The number of processing levels in biological systems has evolved to be optimal for a huge variety of visual recognition tasks and the local inter-level connectivity, with its implicit relative spatial information 'binds' features or parts together at the various levels, to help ensure unambiguous assembly of parts for reliable object recognition at the highest level (Wallis and Rolls, 1997). These assemblies could be considered as biological vision's 'intermediate words'.

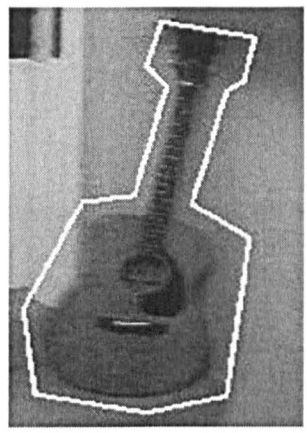


Figure 1.1: Outlining an object of interest for the system to learn to represent
From Johnson and Sugisaka, 2006, Figure 1.

Investigation of the machine vision research in Chapter 3 of the thesis reveals that machine vision systems generally fix the number of processing levels and how they are interconnected, and use this same architecture for all the required object recognition tasks. Feature extraction is designed to capture pre-specified information about the input images, for example to detect the edges of objects. Also, feature selection to reduce combinatorial and high dimensionality problems is conducted on the basis of actively applying user-designed algorithms to determine which features are relevant, or which are irrelevant and should be eliminated.

These findings, coupled with what has been learned in Chapter 2 of the thesis about representation in biological vision, have prompted the formulation of the research questions, which are derived at the end of Chapter 3 and are stated in Section 1.3 below.

1.3 The research questions

There are four research questions posed in the thesis:

Question 1. Is there a general architecture for representing multilevel systems, the same 'formula' being appropriate for a wide variety of representation/recognition problems?

Question 2. Can such systems be self-forming?

Question 3. How can systems find their own descriptors?

Question 4. Is there a way that structure at higher levels can 'emerge' so that the intermediate word problem and the combinatorial and dimensionality problems can be solved automatically?

The thesis attempts to answer these questions through a 'hypothesize and test' approach in a variety of object recognition tasks, including classification of simple geometric shapes, pedestrian recognition and recognition of hand-written numerals.

1.4 The approach taken in the research

The problem of adaptability and the associated issues introduced above are tackled in the thesis through the mathematics of *multilevel hypernetworks* (Johnson, 2006). They naturally facilitate the development of multilevel systems because relations defined on sets of entities at one level give rise to new entities at the next level

A *network* is a mathematical structure that can represent relationships between *pairs* of things from a set. A *hypernetwork* is the generalization of a network to be able to represent relationships among *multiple* things. This provides a way for sets of 'features' or parts at one level to be assembled to form a more complex 'feature' at the next., which relates to the multilevel nature of biological vision. Applying this principle to entities at successive levels can create a multilevel representational framework for object recognition.

The thesis investigates whether *hypernetworks* can be used to construct a general architecture for visual object recognition tasks, as asked in Question 1 above. This hypothesis is tested by constructing *hypernetwork*-based representations for use in recognition of simple geometric shapes, pedestrian recognition and hand-written numeral recognition tasks.

Within a *hypernetwork* framework objects can be represented in terms of their constituent parts and alternatively, the *dual* representation is that of parts expressed in terms of the objects in which they appear. These dual representations can be expressed using mathematical structures called *simplices*. The thesis, in relation to Question 2, explores whether a machine vision

system can use these *simplex* structures to form its own architecture in response to a visual problem.

It is often the case that a part or subset of parts is shared by multiple objects, not necessarily of the same class. The thesis explores how analysis of information about shared parts or features can reveal potential intermediate-level structure for use in discriminating object classes, thus providing a possible approach to answering Question 4.

The issue, raised in Question 3, of how systems might autonomously find suitable low-level representations is explored in the thesis in two ways: through random extraction of features that detect simple patterns of 'light' and 'dark', for recognition of simple, hand-drawn, geometric shapes; and by means of an algorithm that finds image regions that are 'mixture' of 'light' and 'dark', in pedestrian recognition and hand-written numeral recognition tasks.

1.5 Outline of the thesis

After this introduction, various theories of how biological vision contends with the above representational issues are discussed in Chapter 2.

Different approaches of machine vision research into these problems are investigated in Chapter 3. Strengths and failings of some of these approaches are highlighted and through this discussion, the research questions, introduced above, are formulated.

The approaches of the thesis in tackling the research questions in the context of the above issues are introduced in Chapter 4.

In chapter 5, a series of five experiments, designed to address the research questions through the application of various techniques within the framework of hypernetworks theory and in the context of different visual problems, are documented and the results analysed.

Chapter 6 draws conclusions about the extent to which the approaches adopted in the experiments have addressed the research questions, evaluates the contributions of the thesis, and also looks at possible further research in response to new questions arising from the research.

Chapter 2 : Biological Vision

2.1 Introduction

As noted in Chapter 1, biological vision has evolved to enable the individual to cope with the continually changing demands of the environment. To accommodate this ability to adapt, a particular type of multilevel architecture has evolved in the genotype which can be modified in the phenotype by pertinent visual experience throughout life.

Therefore, before undertaking research in machine vision, it is important to try to identify and understand some of the principles of biological vision that could be applicable in artificial systems.

The human visual system is capable of recognizing a wide variety of objects in varying amounts of detail very efficiently, often regardless of position on the retina, illumination conditions, size, viewing angle, surrounding clutter and partial occlusion.

There are thought to be two main visual pathways in the primate brain: the *dorsal* stream, which is concerned with motion and with locating and interacting with objects in the environment; and the *ventral* stream, which is responsible for the representation and processing of information about objects, such as shape, colour and texture, that facilitate object perception and recognition (Rolls and Deco, 2002, p58). There are different theories of how all this is achieved, based on evidence from various disciplines including neurobiological, psychobiological, psychophysical and computational fields. The emphasis of this chapter is on theories of shape representation in *ventral* visual cortex.

Section 2.2 provides a broad overview of the visual system, and in Section 2.3, the feature-hierarchy-based architecture is discussed. Section 2.4 presents various models of object recognition, including feature-based, reconstructionist and holistic approaches, while Section 2.5 covers 3D view-independent versus 2D view-based representation. In Section 2.6 the function of feedback connectivity in hierarchical systems is discussed. The question of the

degree of sparseness of activation within neural populations responding to stimuli is addressed in Section 2.7, while Section 2.8 is concerned with visual perception as inference. Section 2.9 deals with theories of feature binding and selective visual attention, with Section 2.10 concluding the Chapter.

2.2 Overview of the visual system

2.2.1 The Eye

The all important 'front-end' of any biological visual system is the eye, with its photoreceptor cells forming a light-sensitive layer from which nerve fibres emanate to relay information to the brain.

There are two main types of eye in the animal kingdom: simple and compound. The term simple means that the incoming light is received through a single opening, as opposed to sometimes tens of thousands of inputs in the compound eye. The compound eye is made up of a configuration of units called ommatidia, each one of which is an individual light receptor. An ommatidium is composed of a lens, a crystalline cone, an array of light-sensitive cells arranged like the segments of an orange and pigment cells that separate it from its neighbours, Figure 2.1.

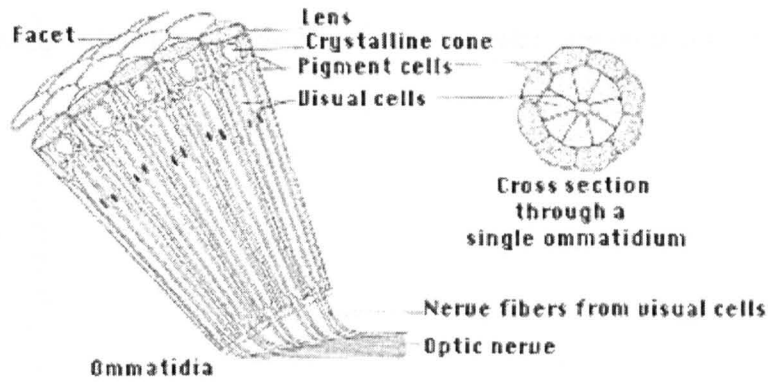


Figure 2.1. The compound eye (Kimball, 2010)

Figure 2.2 shows the faceted eye of the common fruit fly. The responses of all the ommatidia combine to produce a mosaic pattern of light and dark dots, the higher the resolution of which, the better the quality of the resulting image. However, the compound eye is not the best design

for discerning fine detail, providing only about 1/60th of the resolution of human vision. Its strength is in detecting motion. As an object crosses the visual field, the ommatidia that are in line with its trajectory are successively turned on and off, producing a 'flicker' effect. Thus insects tend to be much more responsive to moving than non-moving objects.

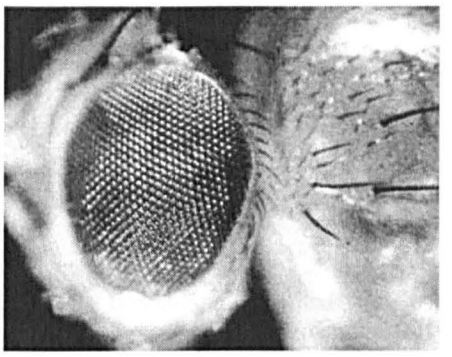


Figure 2.2: The faceted eye of the common fruit fly
Kimball, 2010

The term simple to describe the simple eye is misleading, as, in fact, the construction and functioning of this type of eye can be very complex. Figure 2.3 depicts the various stages of simple eye complexity in molluscs, ranging from a simple pigment spot, through to the optic cup of Nautilus, and the appearance of a primitive lens found in some marine snails, right up to the much more complex octopus eye that has a refractive lens, an iris and a cornea. According to a theory of the Swedish biologist Dan-E Nilsson (Nilsson and Pelger, 1994), these stages may mark some of the developments that have possibly characterized the evolution of the simple eye to the level of complexity of that of the primates, over a relatively short timescale of about 500,000 years.

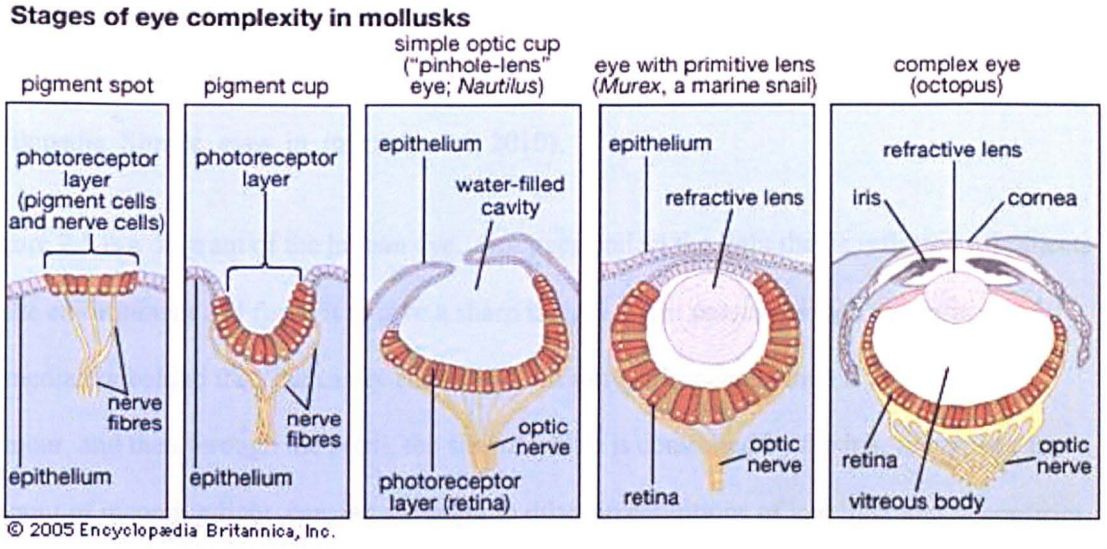


Figure 2.3: Increasing complexity in the simple eye
Encyclopaedia Britannica, 2010

Arthropods such as jellyfish, sea stars and flatworms have pigment spot ocelli (little eyes), with randomly distributed pigment. Spiders have four pairs of pigment cup ocelli (Ruppert *et al.*, 2004), three of which are limited to detecting light direction, but the main pair at the front of the head can form good images, which help in tasks such as hunting or jumping. Many insects, such as bees and wasps have both simple and compound eyes, supported by two anatomically separate visual systems with different functions. In Figure 2.4, three simple eyes can be seen on top of a wasp's head, with a compound eye on either side.

Figure 2.4: Front view of a wasp's head showing simple & compound eyes
Wikipedia_Ocelli, 2010

The ocelli that form the triangular configuration in the picture consist of a small lens and a set of pigmented retinal cells, and are very sensitive to low levels of light and to changes in light

intensity. It is therefore thought that part of their function is to allow the insect to detect the horizon to aid stable flight, and to sense length of daylight to regulate life-cycle (Wikipedia_Simple_eyes_in_invertebrates, 2010).

Figure 2.5 is a diagram of the human eye. Our eyes collect the light that is reflected off objects in the environment and focus it to give a sharp image. Light passes through the cornea and immediately behind that, the cavity containing a watery substance known is the aqueous humour, and then through the pupil, the size of which is controlled by the iris, to regulate the amount of incoming light, causing the pupil to dilate in conditions of low light and to constrict in bright light. The light passes next through the lens and then the vitreous gel that fills the eyeball, and finally hits the retina, the layer of over 100 million light sensitive photoreceptor cells at the back of the eye. The cornea and lens together focus the light, with the lens adapting its shape by means of the ciliary muscles. The lens is made thicker by contraction of the ciliary muscles to accommodate near objects, and thinner by relaxation of the ciliary muscles to accommodate distant objects. Six muscles support each eye in its socket and enable the eyes to move in relation to each other to change depth of focus for near and distant objects (Palmer, 1999, p26). They also permit the eyes to move to fix an object of attention on the fovea, without having to turn the head.

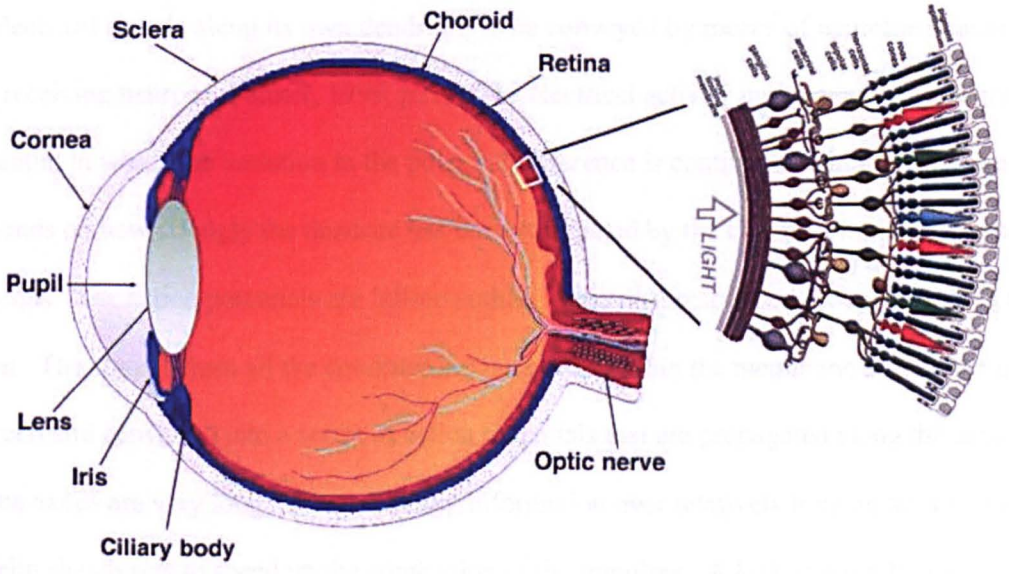


Figure 2.5: Schematic section through the human eye
with schematic enlargement of the retina (Kolb, H. *et al.*, 2010)

2.2.2 The brain

The main processing elements in the brain are the nerve cells or neurons.

An individual neuron generally consists of a cell body, the axon for transmitting signals from the cell to other neurons and dendrites through which the cell receives incoming information from other neurons, Figure 2.6.

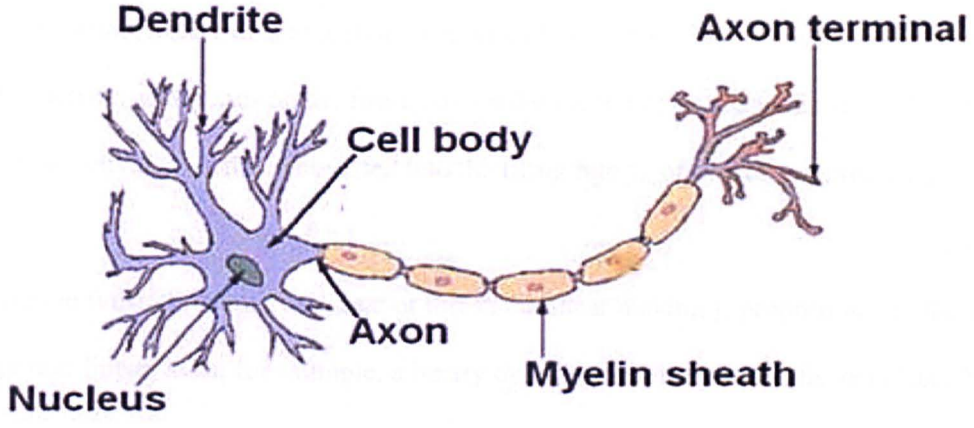


Figure 2.6: Typical neuron
Adapted from Jarosz, 2010

As shown in the figure, the axon terminates in many branches. These terminals form synaptic links with dendrites of other cells allowing the action potentials emitted by the cell in response to electrical signals along its own dendrites, to be conveyed by means of neurotransmitters to the receiving neurons (Palmer, 1999, p29). The electrical activity in the dendrites is a graded potential in which the variation in the potential difference is continuous within a range and depends on how strongly the dendrite has been stimulated by the chemical input from other neurons. The action potentials are 'all-or-nothing' electrical responses, or 'spikes' along the axon. The signals from all the dendrites are integrated within the membrane around the body of the cell and converted into a series of action potentials that are propagated along the axon. Some axons are very long, communicating information over relatively long distances. The myelin sheath acts to speed up the conduction of the impulses. A high spiking frequency, or firing rate, indicates a strong neural response. At an axon terminal, the electrical signal is converted into a chemical one and the resulting neurotransmitter is released into the gap or

synapse between the terminals and the dendrite of the receiving neuron. The stronger the signal, the more neurotransmitter is produced.

The computational equivalent of this process is that a neuron makes a linear weighted sum of its inputs which is its 'activation'. This activation is often denoted as h_i and can be written as

$$h_i = \Sigma_j x_j w_{ij} \qquad (2.1)$$

where x_j is the jth input, Σ_j is the sum of all the input axons and w_{ij} is the strength of the synapse from incoming axon j to the receiving dendrite of neuron x_i The formula shows that the strength of the activation depends on the firing rate on the incoming axon and the strength of the synapse w_{ij}. This activation is then converted into the firing rate y_i, of neuron i expressed as

$$y_i = f(h_i) \qquad (2.2)$$

where the function f(.) can be linear or threshold linear making y_i proportional to the activation h_i or non-linear, as in, for example, a binary or a sigmoid function (Rolls and Deco, 2002, p3).

Figure 2.7 shows conventional notation for an individual model neuron.

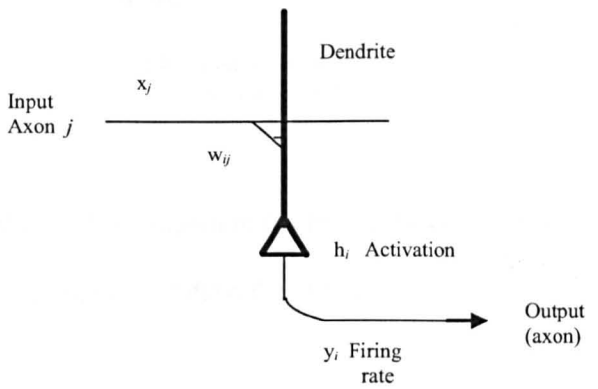


Figure 2.7: Model neuron
after Rolls and Deco, 2002, p4, Figure 1.2

A neuron learns to give a certain response to a particular input through modification to the synapses. In the computational model shown in Figure 2.7, changes are made to the synaptic weights w_{ij} according to a simple 'Hebbian' learning rule of the form

$$\delta w_{ij} = \alpha y_i x_j \qquad (2.3)$$

where δw_{ij} is the change in the value of the synaptic weight w_{ij} in proportion to the input and output firing rates x_j and y_i respectively and α is the learning constant that specifies the amount of change. The Hebb rule suggests that both pre- and post-synaptic firing must occur approximately simultaneously for alteration to the weight to occur.

There is anatomical and neurological evidence that processing within the visual system is divided, broadly speaking, among several 'specialist' areas, (Rolls and Deco, 2002, p36) as seen in Figure 2.8 below. There is some overlap of function in areas that are connected to one another.

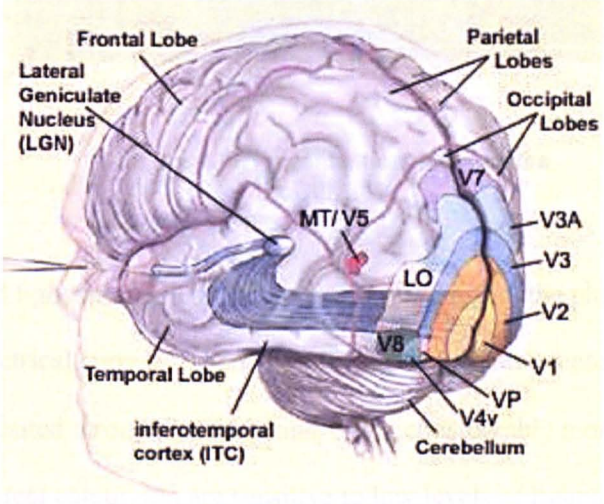


Figure 2.8: Vision in the brain
McGill, 2010

The areas V1, V2, V4 and ITC, shown in the Figure, largely comprise the ventral visual stream and the function of each of them is described in Section 2.3.

The first part of the brain to process visual input is the retina at the back of the eye, Figure 2.9.

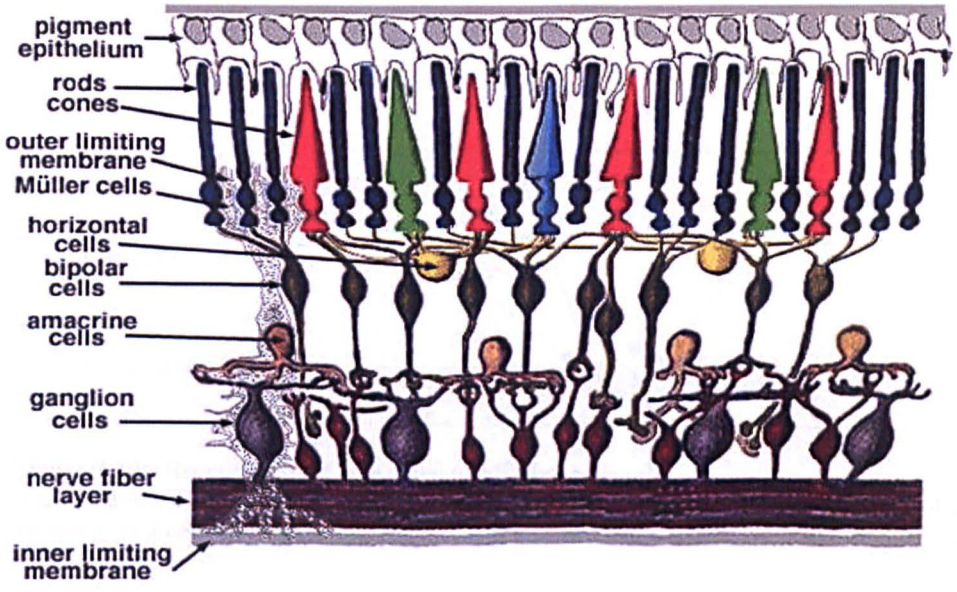


Figure 2.9: Organization of the retina
McGill, 2010

When reflected light from objects enters the eye, it triggers the photoreceptors in the retina to produce an electrical current. There are two types of photoreceptor:

Rods are distributed throughout the retina, being considerably more numerous in the periphery. They do not detect colour, but are sensitive to low levels of light and are slow to respond to changes in brightness. Hence they are mainly used in dark conditions.

Cones, which are less numerous than the rods, are sensitive to colour – red, green and blue - and have their highest concentration at the fovea. They respond much more quickly than the rods to stimulation, being activated by high levels of brightness (Rolls and Deco, 2002, p37).

The photoreceptors send their output to two main types of cell within the retina – horizontal cells and bipolar cells. Some connections run directly from the photoreceptors to the bipolar cells while others take the indirect path, communicating with the bipolar cells through the horizontal cells. This type of direct/indirect connectivity, with a small central region of direct connections surrounded by an area of weaker, indirect connections, forms the receptive field of a bipolar cell. Some bipolar cells have excitatory direct connections and inhibitory indirect connections while others have the opposite configuration, Figure 2.10.

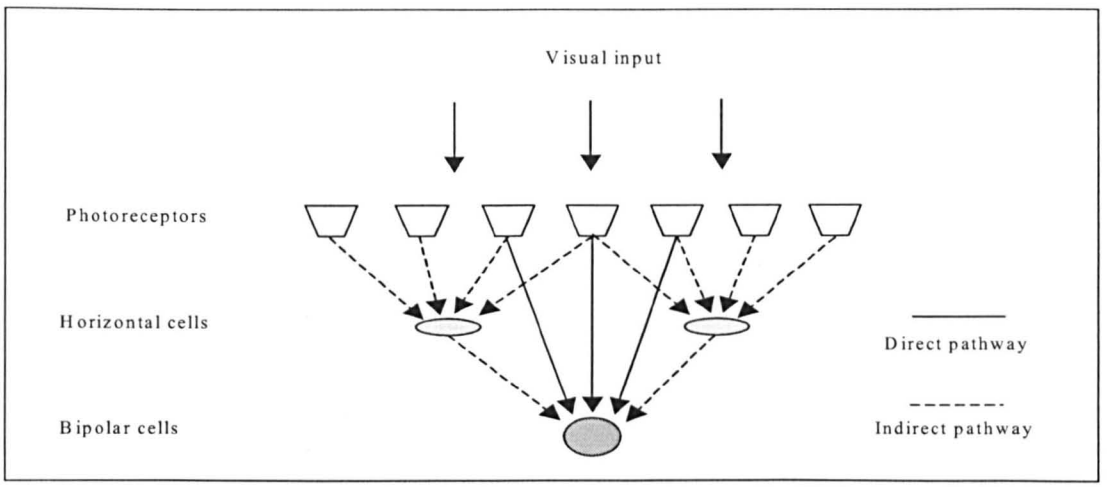


Figure 2.10: Receptive field structure of bipolar cells
with the direct connections forming the centre and indirect connections of opposite polarity coming via the horizontal cells to form the surround, adapted from Rolls and Deco, 2002, Figure 2.3, p39

The output from the bipolar cells is processed by the ganglion cells, the axons of which form the optic nerve. The receptive fields of these cells also exhibit centre-surround antagonism. On-centre/Off-surround cells are maximally activated by a spot of light just large enough to illuminate the centre but not any of the surround, with the firing rate decreasing according to variation in the proportion of light falling on centre and surround, while Off-centre/On-surround cells exhibit the opposite response pattern, Figure 2.11.

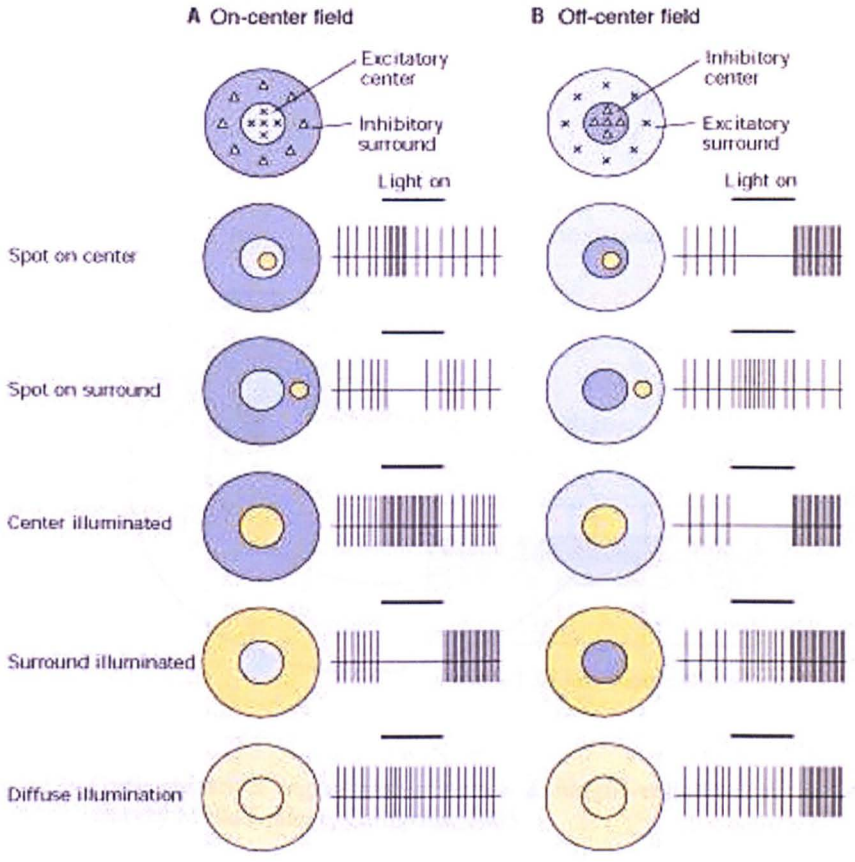


Figure 2.11: The response patterns of the retinal ganglion cells
University of Illinois at Urbana Champaign, 2005

Two different types of retinal ganglion cells have been found:

P (Parvo - small) ganglion cells that receive their input from the cones and are therefore responsive to colour and

M (Magno – large) ganglion cells that are connected to the rods and are thus not colour-sensitive (Rolls and Deco, 2002, p40).

The optic nerves emerging from the retinas first pass through the optic chiasm, where the input from the right visual hemifield crosses over for processing in the left Lateral Geniculate Nucleus (LGN) and the input form the left visual hemifield is routed towards the right LGN. The primate LGN is retinotopically organized with neighbouring regions on the retina projecting to neighbouring regions in LGN, and has six layers, two ventral magnocellular layers and four dorsal parvocellular layers, Figure 2.12.

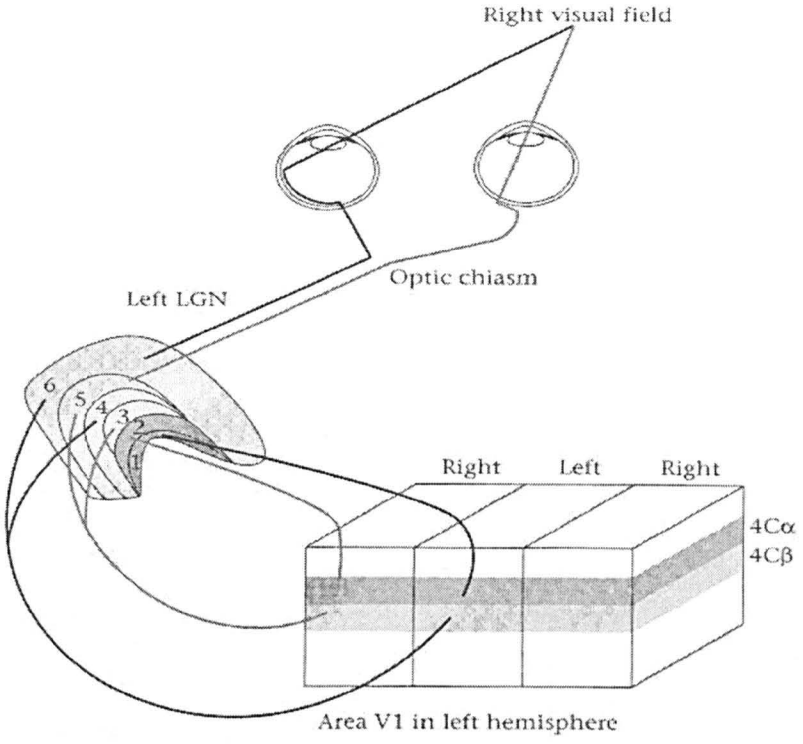


Figure 2.12: Optic chiasm & segregation of parvo- & magno-cellular input to left LGN
Harvey Mudd College, California, 2005

Both parvo and magnoneurons in LGN have circular receptive fields, mostly with centre-surround antagonism. They both have larger receptive fields the further they are from the fovea, but in any given location, magno neurons have larger receptive fields than parvo neurons. Parvo cells respond to colour contrast and give a relatively sustained response to stimuli, making them suitable for processing shape information, while magno cells have greater contrast sensitivity than parvo cells, and they also have much faster and more transient responses, making them better at processing motion and changes in light.

These different characteristics give rise to the two main visual pathways from the retina and on into the visual cortex, Figure 2.13:

The parvocellular system, sometimes called the 'what' pathway, is the ventral visual stream and is concerned with colour and shape processing. After the retinal and LGN stages, colour is processed through the 'blob' subroute, through V1 blob, V2 thin stripe, V4 and on to the

inferotemporal area, while form is processed by the 'interblob' subsystem in V1 interblob, V2 pale interstripe, V4 and finally, inferotemporal.

The magnocellular system, sometimes termed the 'where' pathway, but might be more accurately described as the motion and binocular pathway, is the dorsal stream and after the retinal and LGN stages, it runs through V1 4c α and 4B, V2 thick stripe, V3 , MT and posterior parietal area (Rolls and Deco, 2002, Figure 2.1, p37 and Palmer, 1999, p195).

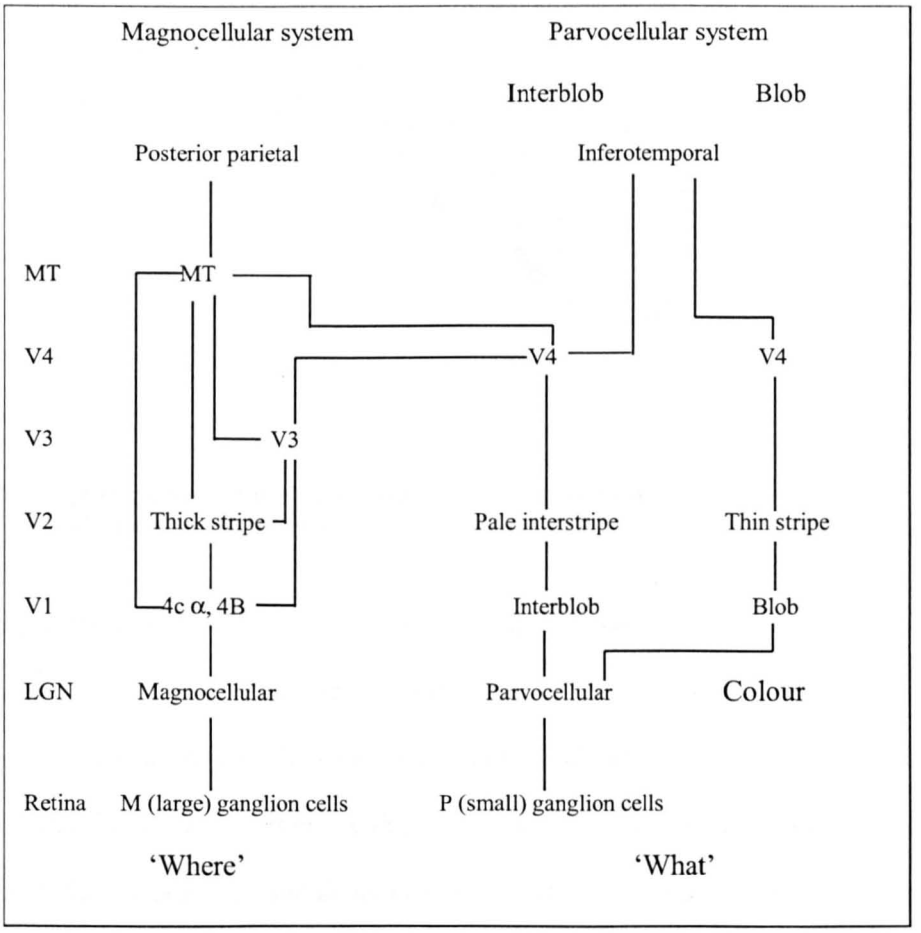


Figure 2.13: The magnocellular and parvocellular pathways from the retina to the visual cortex
After Rolls and Deco, 2002, Figure 2.1, p37

Figure 2.14 shows the approximate paths of the two major visual processing streams in the macaque brain. In addition, a small percentage of the output from the retina is processed along a path through the superior colliculus (SC) in the brain stem, the pulvinar in the thalamus and on to posterior parietal cortex. The superior colliculus is thought to be involved in the integration of sensory information, including visual, auditory and somatosensory input, into motor signals

that guide the head in turning toward the source of a stimulus (Wisconsin University, Neuro-anatomy, 2010), while the pulvinar also appears to integrate multiple sensory input as well as being involved in eye movement (Best, 2010).

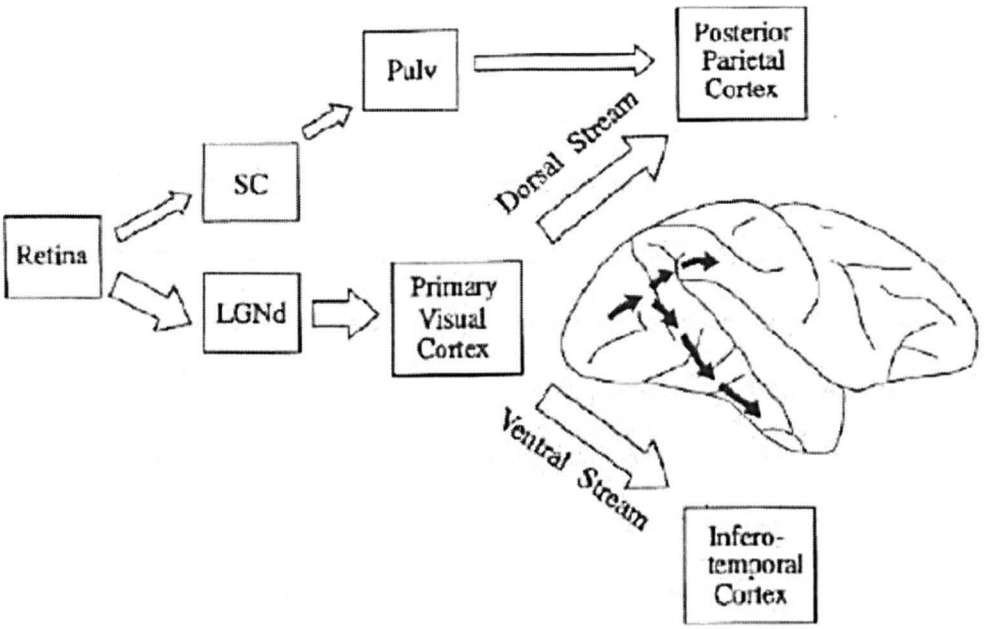


Figure 2.14: Approximate paths of the dorsal and ventral streams in the macaque brain
Milner and Goodale, 1998

The macaque brain diagram on the right shows the approximate routes of the two main pathways from the primary visual cortex to the posterior parietal and the inferotemporal cortex, respectively. The path through the superior colliculus and the pulvinar and on up to the posterior parietal cortex is concerned with processing mulit-sensory information about where objects are in the surroundings and about eye- and head-movements – (adapted from Palmer, 1999, p35). (LGNd: lateral geniculate nucleus, pars dorsalis; Pulv: pulvinar; SC: superior colliculus).

The primary visual cortex, or V1, is the largest primate cortical area, covering about a third of the neocortex in humans. It has six functionally distinct layers, labelled one to six from the outer surface inwards with LGN feeding into layer 4, from which neurons project to visual region V2 (Rolls and Deco, 2002, p48). It contains about 200 million neurons as compared to

LGN's 1.5 million and the retina's 1 million ganglion cells (Palmer, 1999, p151), and largely preserves the retinotopic organization in LGN.

V1 neurons are selective for different aspects of the input including location, scale, orientation, colour and motion, as well as for which eye the signal originates from (Rolls and Deco, 2002, p45). Neurons selective for one eye more than the other tend to group together into what are termed 'ocular dominance slabs', long thin slices of tissue that run parallel to the cortical surface and are interleaved in a modular arrangement of 'hypercolumns' running perpendicular to the cortical surface. Every retinal location is represented by its own hypercolumn in V1, providing coverage of the input from each eye, with a progression of cells tuned to different orientations along one dimension and possibly to different spatial frequencies running perpendicular to the orientation dimension (Palmer, 1999 p157, Figure 4.1.15 and Rolls and Deco, p47, Figure 2.9). The segregation of the ocular input from LGN occurs in layer 4 of V1, with the neurons in the other layers of each hypercolumn taking input from both eyes allowing for perception of depth through binocular disparity. In addition, at intervals along the orientation dimension, in cortical layers 2 and 3, there are 'blobs' containing cells that respond to colour.

2.3 The feature hierarchy theory

Hubel and Wiesel (1962), (Hubel, 1995), proposed that the visual system is hierarchical, based on their findings relating to the preferred stimuli of neurons in primary visual cortex. They found cells in V1, that respond to bars at a particular orientation and position within their small receptive fields and other cells that are sensitive to specifically oriented bars occurring anywhere in their larger receptive fields and that thus respond to a moving stimulus being swept across their receptive field. Hubel and Wiesel termed these cells 'simple' and 'complex' respectively and postulated that each type of cell receives its input from several lower-level cells with overlapping receptive fields. Figure 2.15 illustrates the possible construction of a simple cell's elongated receptive field from overlapping circularly symmetric receptive fields of LGN

neurons and Figure 2.16 shows how a suitable configuration of the receptive fields of simple cells could give rise to the characteristic responses of a complex cell.

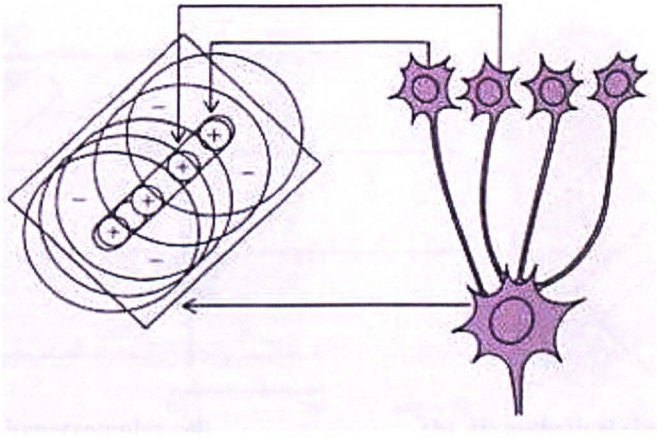


Figure 2.15: Possible formation of a simple cell's elongated receptive field
"Possible connections from several overlapping cells in LGN, say, could form the receptive field of a simple cell. Many more such cells would be required than are shown in the figure. In this example, the lower-level cells are On-centre Off-surround and together they form a receptive field with an elongated excitatory centre and inhibitory regions on each side. This simple cell would respond maximally to a long narrow slit of light falling on the central strip of its receptive field, at the appropriate orientation.", from Hubel, 1995, p15 of Ch 4.

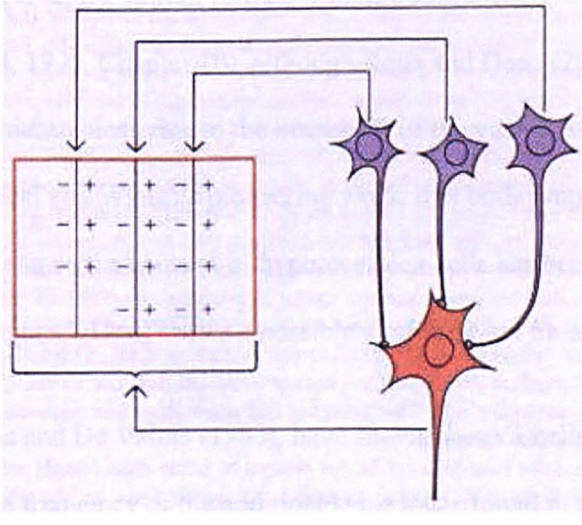
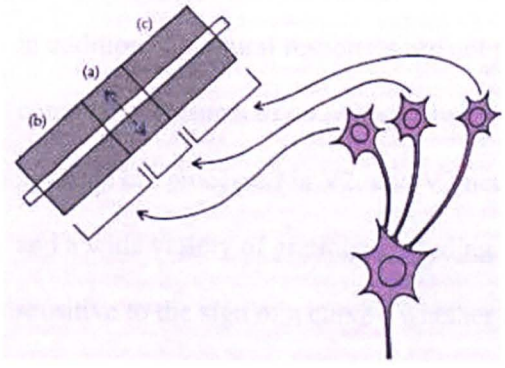


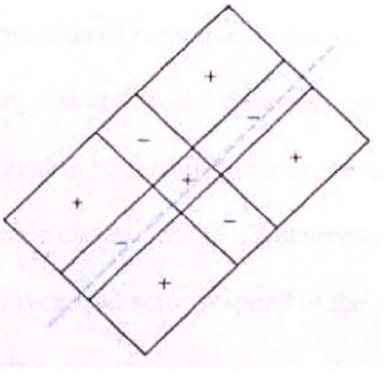
Figure 2.16: Possible formation of a complex cell's shift-tolerant receptive field
How several simple cells, with, in this example, selectivity for vertically oriented stimuli with light on the right side, could provide for the characteristics of a complex cell that responds to a vertical edge appearing anywhere within its receptive field, but due to the process of adaptation, only has a sustained response to a moving stimulus.

When Hubel and Wiesel found end-stopped cells, that respond optimally to appropriately oriented bars of a certain length, but are suppressed by bars exceeding that limit, they termed

them hypercomplex cells because they assumed that these neurons were at a later processing stage than the complex cells and received inputs from them, as shown in Figure 2.17(a).



(a): Hypothetical hypercomplex cell
taking input from three complex cells, with cell (a) providing excitatory input, while cells (b) and (c) are inhibitory



(b) Hypothetical simple end-stopped cell
receiving, in this example, central excitatory connections with inhibitory inputs on either side from ordinary simple cells

Figure 2.17: (a) Hypothetical hypercomplex cell with inputs from three cells; (b) Hypothetical simple end-stopped cell
(after Hubel, 1995, Ch 4, p23 and p24)

They postulated that the visual system is hierarchically organized with each successive visual area processing more complex stimuli formed from the conjunction of simpler inputs from the previous level (Hubel, 1995, Chapter 10), although Rolls and Deco (2002), point out that the system is not totally hierarchical, due to the branching of the ventral and dorsal streams. It has been found, since Hubel and Wiesel's pioneering work, that both simple and complex cells can exhibit endstopping, and so the concept of hypercomplex cells has been dropped (Rolls and Deco, 2002, p43). Figure 2.17(b) shows a possible configuration for a simple end-stopped cell.

In addition, De Valois and De Valois (1988), have shown that V1 cells are sensitive to spatial frequency through the frequency of the additional side lobes found in the response profile of their receptive fields (Rolls and Deco, 2002, p43). Furthermore, V1 cells may have complex shape preferences, possibly arising from interactions of the non-classical surround (Das and Gilbert, 1999), or from complex selectivity within their classical receptive field (Hegde and Van Essen, 2000).

V2 cells are retinotopically arranged, with larger receptive fields than those in V1 and much of their forward input coming in the form of combinations of responses from subsets of V1 neurons. As a consequence, they tend to respond to more complex stimuli than V1 neurons, and in addition, V2 neural responses are not predictable by summation of responses to the individual component features of complex stimuli. Colour, orientation, size and spatial frequency, as well as shape are processed in V2, and V2 neurons have been found to be sensitive to arcs, circles and a wide variety of gratings, including spirals and concentric circles. Some V2 neurons are sensitive to the sign of a curve - whether it is concave or convex, and some respond to the polarity of an angle – whether it is acute or obtuse. V2 neurons can also be responsive to illusory contours and border ownership (Hegde and Van Essen, 2000).

Many V2 cells respond to illusory contours as though they were contrast borders. They respond to illusory moving bars, Figure 2.18 below (from Von Der Heydt, 2003, Figure 2, p16), and also to illusory contours, for example, between abutting grating patterns at the preferred orientation for a real moving bar, Figure 2.19 below (from Von Der Heydt, 2003, Figure 3, p17).

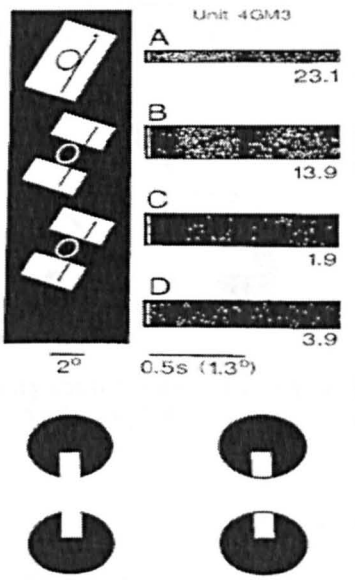


Figure 2.18: V2 cell response to illusory moving bars
from Von Der Heydt, 2003, Figure 2, p16.

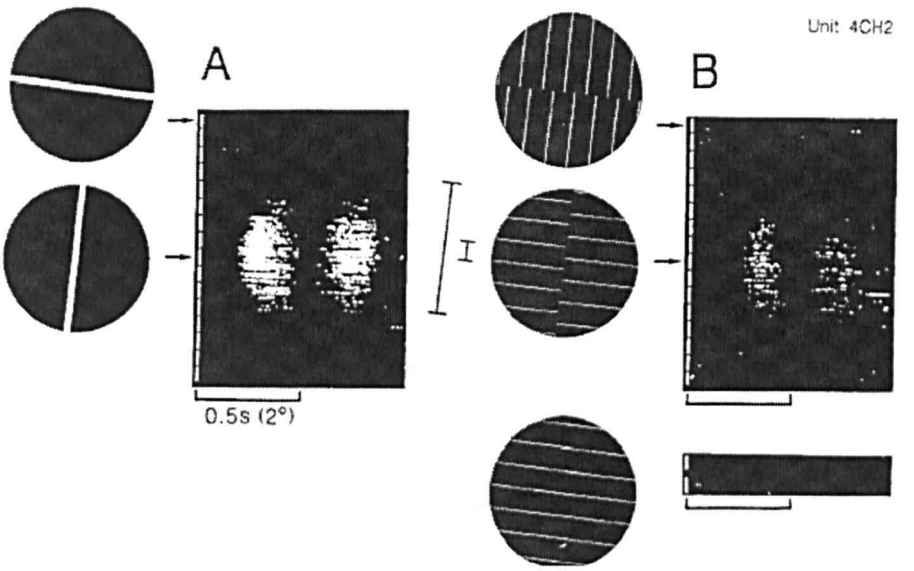


Figure 2.19: V2 cell response to illusory contours
from Von Der Heydt, 2003, Figure 3, p17.

Their response seems to take into account information from outwith the classical receptive field (CRF). In the perception of illusory contours, an occluding figure is assumed and as a result, its contours are 'created'. The occluding surface then appears brighter than the background, Figure 2.20 below (from Von Der Heydt, 2003, Figure 1, p15).

Figure 2.20: Illusory contours give rise to a 'bright' occluding shape
from Der Heydt, 2003, Figure 1, p15.

A similar effect occurs where there are borders of sharp contrast. These borders are again perceived as being occluding.

Some V2 neurons exhibit side-of-figure preference in border detection, in conjunction with orientation and colour selectivity, for example. V2 cells can also respond to edges defined by disparity, as in a random-dot stereogram, and they also have a preference for a particular depth

edge, depending whether the occluding surface is to the right or to the left. It is thought that border ownership is represented in the difference in the responses of pairs of neurons with the same orientation selectivity, but opposite side-preference, Figure 2.21 below (from Von Der Heydt, 2003, Figure 9, p23).

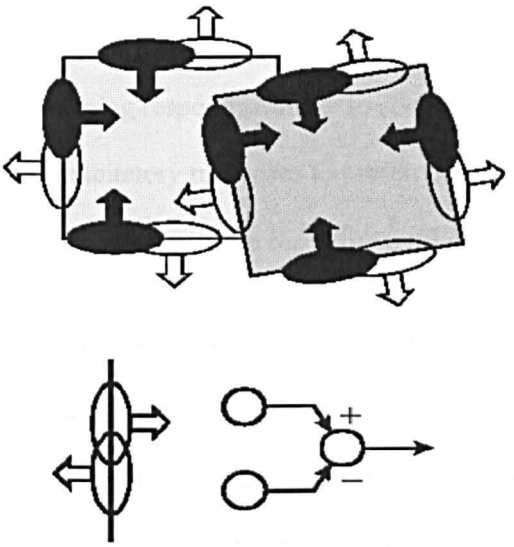


Figure 2.21: Opposite side preference of pairs of V2 cells establishing border ownership from Von Der Heydt, 2003, Figure 9, p23.

Von Der Heydt (2003) suggests that the early visual areas, especially V2, are generating image-parsing hypotheses that can be assessed by higher cortical areas. This lends weight to Wolfe's extension to his Guided Search Model (Wolfe, 1996) to include parsing of images with overlapping shapes and also fits with Lee's 'high resolution buffer hypothesis' (Lee, 2003). However, he goes on to point out that since the receptive fields in V2 are relatively small, the contextual information necessary for encoding border ownership has to be provided from outwith the CRF, possibly through lateral connections within V2, or alternatively V4 may provide more global information to V2 via back-projection (Zhou *et al.*, 2000).

V4 is less retinotopically organized than V1 and V2, and V4 neurons have quite large receptive fields relative to those in preceding cortical areas. They receive overlapping input mostly from V2 and have overlapping receptive fields themselves. They are also generally more broadly tuned for orientation and spatial frequency than are the early visual areas. Pollen *et al.*'s model (Pollen *et al.*, 2002) assumes that the receptive fields of V4 neurons are comprised of subfields

that each receive their input locally from earlier cortical areas and that V4 responses to preferred simple stimuli are invariant except in the degree of activation in different regions of the receptive field. They have found, through subfield analysis, that V4 neurons seem to respond most to single common values of orientation and spatial frequency across the whole receptive field. Some cells show virtually no length- or width-stopping, while others show different degrees of length- or width-stopping or both. Pollen *et al.*'s (2002) results show that V4 neurons that exhibit end-stopping respond strongly to complex stimuli, such as concentric gratings, that both enhance excitatory responses to preferred orientation and spatial frequency, and reduce inhibitory interactions across the receptive field.

Several studies, including that of Pasupathy and Connor (1999), have found that V4 neurons participate in coding curvature, with some V4 neurons responding better to contour features than to simple edges or bars. In later work, Pasupathy and Connor (2001) show that some V4 neurons seem to encode fairly complex information - concavity/convexity and the polarity of angles - about sections of boundary in particular locations - for example, top right – within shapes, in relation to their centre of mass. V4 neurons also seem to be involved in computing border ownership. As well as in V2, Von Der Heydt (2003) has found cells in V4 that are side-of-figure selective.

IT neurons receive most of their input from V4. By a process of gradual reduction in the complexity of stimuli presented, Tanaka and colleagues (Tanaka, 2003) found that IT neurons generally respond to moderately complex subsets of features, rather than to whole objects, and hence several IT cells are generally required to represent a complete object. However, there are face-selective IT neurons that only respond when virtually all the important facial features are present and some need the features in the correct arrangement (Rolls and Deco, 2002).

IT cells also respond to shading information, given an assumed direction of the light source, and also to stereoscopic depth cues, which suggests they can be sensitive to implied depth as well as to 2D shapes, however, it is thought that IT does not fully recreate 3D structure, but that the disparity and curvature information simply enhances the 2D representation (Tanaka, 2003).

Tanaka and colleagues (2003) have found that preference for a particular shape is maintained across the receptive fields of IT neurons, with the maximum response occurring around the centre and decreasing gradually towards the periphery, thus providing some spatial information. Also their receptive field centres tend to be arranged around the fovea, providing detailed representation of shape, colour and texture.

Scale invariance has been found among IT neurons. It may be that scale-specific cells feed into a receiving cell enabling it to respond to the appropriate shape regardless of size, or alternatively, it may be that scale-dependence and scale–independence both operate in IT. A number of other types of invariance have been found in IT, including response to contrast reversal, changes in lighting, direction of motion, coarseness of texture and so on. In addition, some IT neurons have been found to insensitive to changes in aspect ratio of the sort that occur when an object is rotated in depth.

Also, Tanaka and co-workers have found a columnar arrangement of selectivity to similar stimuli, perpendicular to the cortical surface, while similar selectivity only extends for a short span oblique to the cortical surface. Partial overlapping of columns sensitive to different, but related features was found, especially for faces appearing in different views. In addition, some neurons have been found to be more sensitive to the global configuration of parts, such as vertical alignment, rather than to the characteristics of the parts themselves, which could be useful for 'feature binding' in order to prevent detection of illusory objects.

Groups of neurons with overlapping, but slightly different selectivities could therefore provide view-invariant representation and could allow generalization among similar objects within a particular category. Or, alternatively, subtle differences in a given feature could be represented in the differing activities of a set of cells that respond to slightly different stimuli (Tanaka, 2003).

2.4 *Models of invariant object recognition*

2.4.1 Feature hierarchy-based models of invariant object recognition:

Fukushima's Neocognitron (1980), and its subsequent extensions (including Fukushima, 2004) demonstrate that it is possible to build up a degree of transform invariance by means of a hierarchical system based on the theory of Hubel and Wiesel, applied to hand-written character recognition. The original model consists of eight layers, each of which has a set of simple 'S' cells, which are distortion-, position- and size-dependent, followed by a layer of complex 'C' cells, that receive input from sets of the previous layer's 'S' cells, thus providing some transform invariance. Competition is driven by excitatory connections between layers and inhibitory connections within layers (Satoh *et al.*, 1997). Thus the system derives the ability to generalize over distortion, position, and size to a small degree. However, a non-biological aspect of the system is that, once a pattern has been learned by a particular neuron, through competition in a given layer, that neuron is replicated throughout the layer by a non-local 'learning' mechanism (Rolls and Deco, 2002). The model has been extended to tackle rotated patterns (Satoh *et al.*, 1997) and incremental learning (Fukushima, 2004).

The system devised by Wallis and Rolls (1997) – VisNet, and Rolls and Milward, (2000) – VisNet 2, has a more neurologically plausible learning scheme. The basis for the model is neurophysiological evidence from single-neuron studies in primates, mainly in processing information about faces. Face sensitive neurons were chosen for study since they are readily found, due to their comparatively large numbers in the visual cortex.

The model consists of four hierarchically arranged layers of competitive networks, with mutual inhibition among neighbouring neurons in each layer. The receptive field sizes of the cells increase up through the layers as a result of sets of connections from local cell populations in preceding layers converging onto each cell in the next layer. The system learns transform

invariances by means of a modified Hebb-like rule, incorporating a memory trace of each cell's previous activity. The system has been extended to model the feature-binding problem and attentional mechanisms, discussed in Section 2.9 of the thesis.

A feature hierarchical model based on that of Fukushima, with alternate simple cell and complex cell layers, has been devised by Riesenhuber and Poggio (1999). The network's layers of 'S' cells build increasingly complex features, while the 'C' cells provide some translation invariance as with Fukushima's scheme. The authors argue that, with the linear summation approach to pooling neural response, feature specificity is lost, since the response level is determined by all the afferents, so they propose a non-linear 'MAX' mechanism to be implemented by the 'C' cells in the process of building invariance, whereby the activation is only dependent on the maximum response and so indicates which is the best-matching part of the stimulus to that neuron's preferred feature pattern. The authors point out that this approach could avoid ambiguity in interpreting cluttered scenes or processing multiple objects. The system is hard-wired and in that way is not very biologically plausible, but something akin to the 'MAX' operation may by used in providing invariant object representation in the brain (Rolls and Deco, 2002, Gawne and Martin, 2002).

2.4.2 Correspondence-based models

The approach of Olshausen, Anderson and Van Essen (1993) to translation and scale-invariant object representation employs a set of 'control' neurons to map a particular set of inputs within an attentional window, up through the network to the appropriate output neurons for the amount of translation or scaling required, thus transforming the contents of the attentional window into a position- and scale-invariant object-centred reference frame. The control neurons effect a multiplicative modification of the weights on the afferents to neurons in the system, thus enabling the network to reconfigure its effective connectivity to suit a particular task. Multiplying and remapping significant parts of the retinal input onto a particular set of outputs seems biologically implausible. Also, the control signal for the multiplication has not been located in the brain (Rolls and Deco, 2002). Another interesting point Rolls and Deco make is

that, if the visual system did employ this approach, there would seem to be no need for a hierarchical arrangement, since the invariance problem could theoretically be solved in a single layer. Also it is difficult to see how the control mechanism would be learned in the first place (Mel and Fiser, 2000).

A similar approach is taken by Zhu and von der Malsburg (2004) in their invariant object-recognition model. The assumption is that organized dynamic links are set up in the brain through learning during infancy, and that these organized configurations then serve as the memory traces of ordered mapping patterns for frequently-encountered structural features. This helps to overcome the problem of lengthy convergence times that affect the performance of dynamic link-based systems. The maplets have a similar role to the control neurons in Olshausen *et al.*'s, (1993) model with each maplet representing the relative position, orientation and scale between small regions in the input image and similar regions in the stored model. The subgraphs that represent elements common to many ordered mappings are high-order links forming connections between small regions in the input image and similar regions in the stored 'model'. This enables the system to make within-class discriminations among objects, such as faces, that tend to represented holistically. However, Zhu and von der Malsburg point out that categorization of composite objects belonging to different classes requires more that just one correspondence stage of processing. Such objects have to be represented as ordered sets of parts as, for example, in Biederman's (1987) "Recognition by Components", discussed below.

2.4.3 Reconstructionist and holistic models

Marr (1982) is of the opinion that the functioning of the visual system can not be fully understood by considering neurological evidence alone. Nor is it sufficient to come up with a plausible algorithm that the system might implement in solving the vision problem. The way to understand how vision works, in Marr's view, is to first establish a theory of what the system is trying to compute, which might then inform research into the nature of the subordinate levels of algorithm and neural 'hardware'.

Marr's representation system is hierarchical and 'reconstructionist', with the different stages intended to correspond roughly to the different levels of the visual system:
a 'retinal' input layer of image intensity values; the 'primal sketch' level for the detection of edges, lines, boundaries, blobs and groups etc corresponding to V1; the 21/2 D Sketch, providing local surface orientation, distance from viewer, depth discontinuities in surface orientation information, developing in V2 – V4, with parsing of the resulting reconstructed shape and conversion to the 3D sketch ready for comparison with the set of prototypes in IT (Lee, 2003, Figure 2); This process involves the transformation from a viewer-centred, image-based description to an object-centred reference frame, parsing the shape into volumetric primitives – generalized cones, before comparing it, on the basis of its parts and the syntax required to bind them, with a set of prototype shapes. This approach gives a view-invariant representation of objects.

Marr himself mentions the necessity for prior image segmentation. He also mentions the problem of finding suitable axes for objects such as screwed up paper. In addition, he discusses the difficulty of extracting 3D axes from 2D images of objects appearing at different viewing angles etc. Another problem with this approach as a model for how the visual system achieves object recognition is that it can be difficult to segment objects into parts, particularly non-rigid objects like a cat curled up asleep (Rolls and Deco, 2002). Furthermore, it is difficult to see how a syntactical representation could be implemented neurally. Ensembles of neurons could represent sets of parts, but the problem then would be how to specify the order of linkage of those parts – for instance that the tail should be connected at the rear of the animal, not to its head.

Another reconstructionist model is Biederman's (1987) Recognition by Components (RBC) which derives from Marr's approach, but instead uses a small alphabet of volumetric, generalized cones, termed 'geons', as primitives. The 3D geons can be uniquely specified in terms of their 2D 'non-accidental' properties by the application of certain 3D inferences – and the objects are represented as configurations of geons and their spatial relations, therefore it is not necessary to convert to an object-centred frame of reference. 'Non-accidental properties' is

the expression coined by Biederman for the properties that are generally preserved during conversion from 2D to 3D representations, namely straight lines remain straight, curved lines remain curved, symmetry under rotation and reflection is retained and junctions of lines and parallel edges also remain. The model stresses view-point invariance and the ability to cope with incomplete representations – due to occluded parts. This is more flexible than Marr's requirement that a part is either present or not. Again, the method requires prior segmentation of images. In addition, Biederman talks of parsing objects at concave regions, but this can be difficult to implement and has problems if applied universally for all categories of objects – defining where fingers begin in relation to a hand, for example (Fleck, 1996).

Another strategy for representing objects is by means of a holistic description, within a shape-space spanned by a basis set of vectors each representing a prototype shape (Edelman, 1999). A new shape is then encoded by its similarity to the set of prototypes vectors with which it has any correlation and is represented by a point in the pattern space. The scheme is based on a network of neurons each tuned to a particular view of a given object. However, the tuning is relatively broad so that the system can interpolate to novel views. This approach has the advantage that similar shapes of object tend to group together in the shape space. Of course, objects within a particular class can vary considerably in size and shape, and conversely, objects of very different categories can look rather similar, so any shape-based classification scheme would require a top-down 'override' mechanism for dealing with these cases.

One problem with the holistic approach is that, while humans can describe the similarities and differences between complex objects on the basis of their parts, this system is likely to perceive different configurations of the same parts as indicative of completely different objects. This can lead to the system failing to perceive rotations of objects in the plane – for example a sphere on top of a cube being inverted to give a cube on top of a sphere - whereas a parts-based, structural approach would detect this. Edelman's 'Chorus of Prototypes' system (Edelman, 1999) also suffers from the need for prior image segmentation and the shapes are normalized in size to accommodate the similarity measure.

2.5 Object-based versus view-based descriptions

The above models can be divided into systems that use 3D, object-based, view-invariant representations of objects and those that use 2D, image-based, view-dependent representations. Recognition in the former is by matching of parts and their spatial arrangements and in the latter, by interpolation from previously learned views.

There is considerable disagreement about whether representation in the visual system is view-independent or view-dependent. In terms of psychophysical reasoning, if it is possible to extract an object-centred reference frame regardless of viewpoint, and thus derive a view-invariant representation, then both the speed and accuracy of recognition of an object should be unaffected by the viewpoint of the observer. On the other hand, if the representation is viewpoint dependent, then the speed and accuracy of recognition should be dependent on the magnitude of the transformation required to bring the object into 'register' with a familiar view. However, another approach is to employ a set of several 2D views and recognition is by interpolation among these views rather than in terms of alignment to a single view (Bulthoff *et al.*, 1995).

Various factors might influence the type of representation required (Logothetis and Sheinberg, 1996). The level of description of an object is important, whether it is at the basic level, such as 'cat', 'dog', or at a more specific, identification level termed the subordinate level, 'Mittens' the cat, for example, or at a super-ordinate level, 'mammal', for instance. Models for recognizing different categories of object have tended to use a view-invariant representation scheme (Marr, 1982 and Biederman, 1987), whereas models for recognizing particular instances of a given category of object have used a 2D view-dependent approach (Wallis and Rolls, 1997, faces, and Riesenhuber and Poggio, 1999, paper-clip objects, for example).

An associated factor that affects recognition performance is the level of expertise of the observer since this suggests that the expert observer's entry-point to recognition is likely to be automatically at the subordinate level, for example, being able to describe the object as a 'Peugot 307', rather than just as a 'car'. This subordinate level of recognition is thought to be strongly view-dependent at first, with the ability to generalize increasing with experience. Brain damage seems to be able to affect recognition at the subordinate level without affecting categorization ability, for example in prosopagnosia, in which there is loss of the ability to recognize individual faces at the subordinate level, but not to classify objects at the basic level. Other considerations are the intended action involving the object – say grasping – and the shape and structure of the object, as well as biologically significant objects such as faces, hands and so on. Hence Logothetis and Sheinberg (1996) take the view that there are likely to be multiple specialized visual recognition systems in the brain, in particular, that structural representations tend to be used in object categorization and image-based descriptions for within-class identification – as noted above.

A 3D structural view-independent representation, while it avoids the need for matching-up or alignment processes, has the problem of recovering the appropriate 3D structure from the 2D image, regardless of the viewing angle. Biederman and Gerhardstein (1993) set out three conditions that must be met so that view-point invariant representation of an object can be achieved: the object must be decomposable into view-invariant parts; every object must have a unique description in terms of parts and their relations; the structural description must remain constant over different viewpoints. This implies, and the authors point out, that not all objects fulfil these requirements –crumpled paper, for example, does not decompose into regular parts, and so such objects cannot be described invariantly. In addition, studies designed to satisfy Biederman and Gerhardstein's conditions have shown that human object recognition is largely view-dependent (Hayward, 1998).

However, there are also problems with image-based models. The fact that an image shows an object from a single viewpoint makes generalization from familiar to novel examples of an object category difficult. Without the ability to generalize, a large number of exemplars of each

object type may have to be stored. In addition, some sort of normalization scheme and matching algorithm is needed to generalize among the stored exemplars and their different views. This approach requires prior segmentation of the image and an implied prior recognition of the novel exemplar in order to know what transformation is required to align it with a stored representation. A way round this problem is through interpolation across different views of an object and across different exemplars of a class. A given view is represented by a set of view-dependent features, which, in turn, can be depicted as a point in a multi-dimensional space of all possible views. And in the case of interpolation among different members of a given class, each individual is represented as a set of exemplar-specific features, for example, for faces, the features might be eyes, nose, mouth, chin etc., and again the object is represented as a point in the space of all faces. Edelman's 'Chorus of Fragments' model (Edelman and Intrator, 2002) is similar to this. A novel stimulus is then classified on the basis of its similarity to a number of neighbouring representations.

What seems to be lacking to date is an image-based system that can not only interpolate across views of a single object and across exemplars of a single class, but can interpolate across views of single exemplars of very many different classes. Recognition would involve processing large amounts of information, if individual learned exemplars were stored for each class. One theory (Logothetis and Scheinberg, 1996) is that when the number of exemplars grows large, the representation is on the basis of prototypes, where the 'central tendency' or 'average exemplar' of a class, although never having actually been seen, comes to represent that class.

So the question remains as to whether object recognition is viewpoint invariant under certain circumstances. It has been claimed that recognition delays found with view-dependence apply only to subordinate-level descriptions, implying that general object categorization is view-invariant. However, Tarr and Cheng (2003) point out that delays also occur with entry-level recognition.

The delay and often-associated decrease in recognition accuracy with view-based recognition is assumed to correspond to the degree of mental transformation required for the observer to align

an unfamiliar view of an object with the nearest canonical stored view. The difficulty with this is that the alignment approach has the problem of a large search space of possible transforms (Tarr and Bulthoff, 1998). However, Perrett *et al.* (1998) present evidence for an alternative to mental transformations to explain the delay in view-based recognition. They explain that if the representation is view-based, cells will tend to become tuned to the views that are more commonly experienced, so that the further away from familiar views a newly observed object is, the fewer cells will respond to it and hence the longer it will take for sufficient input to be accumulated for activating the receiving, decision-making neurons. This effect has been found in generalization to unusual viewing conditions in the context of rotation in depth, orientation, scale and occlusion.

2.6 The purpose of feedback in hierarchical visual systems

The models described in Section 2.4 are largely based on a feed-forward architecture. However, visual areas connected feed-forward are also connected through feedback, Section 2.2. The role of feedback has been the subject of much research and has been found to be significant in several important visual functions. For example:

resolving ambiguity in the representation in lower visual areas through top-down inference, based on previous experience or the requirements of the task in hand, as discussed in Section 2.8 of the thesis.

directing attention to salient areas or objects within an image to extract information pertinent to task demands, through competitive mechanisms, Section 2.9.

In learning, back-projection of stored stimuli can be used to either help separate out very similar input patterns, or to group together rather dissimilar inputs for categorization. It can also guide a competitive learning system in what it should learn, through feedback about the significance of input stimuli (Rolls and Deco, 2002, p32).

In recall, when neurons in higher areas have learned to represent more than one mode of stimulus, for example, the sight and the taste of food, if the taste representation is later back-projected, it will cause re-activation of the neurons that were originally activated by the sight of

the food. Evidence that relatively early visual areas are activated during visual recall has been found by Kosslyn (1994, p287) and Rolls and Deco (2002, p34).

Hence it seems unlikely that neurons are merely acting as general 'multi-purpose' filters, as in purely feed-forward processing, but rather as highly-specialized feature detectors with responses shaped by to-down influences as well.

Lee *et al.* (1998) argue that processing in V1 cannot be completed before the computations in high-level areas have begun. Lee *et al.* also present neurophysiological evidence that V1 cells are computing more detailed information than just the oriented edges and bars of early visual processing further through the post-stimulus firing time of duration 40 – 350ms and that they may be providing detailed orientation and spatial information to extra-striate areas via a feed-back loop. The authors hypothesize that V1, and the other early visual areas, LGN and V2 - form a high-resolution buffer, that is involved in many levels of visual processing, being influenced by context, higher-level inference, task demands and previous experience (Lee, 2003).

2.7 Sparseness of representation

Another consideration for object representation in the ventral temporal cortex is the proportion of neurons from a population that are active at one time in any given representation, whether based on features or processes.

When only one neuron out of the ensemble is active in response to a stimulus, this is known as local encoding, with the single active neuron sometimes being termed a 'grandmother cell' to convey the concept of a different single neuron responding to each different object. At the other extreme, a fully-distributed, or dense representation has, on average, half the neurons active at any one time, while a sparse code only involves a small proportion of the neural ensemble to be active in the representation. The type of encoding used affects the information processing capabilities of neural networks with regard to representation and storage capacity,

generalization, speed of learning, error tolerance, control of interference between stored patterns and so on, Table 2.1 (after Foldiak, 2002, Table 1).

	representational capacity	memory capacity	speed of learning	generalization	interference	fault tolerance	simultaneous items
local	Very low	Limited	Very fast	None	None	None	Unlimited
sparse	High	High	Fast	Good	Controlled	High	Several
dense	Very high	Low	Slow	Good	Strong	Very high	one

Table 2.1: Properties of coding schemes
from Foldiak, 2002, Table 1

There is evidence that sparse coding tends to be used in the processing of sensory information in the brain. Local encoding would require an unmanageably large number of neurons to represent all the different stimuli the visual system is likely to encounter, while in a dense coding, there would be much overlap in the representation, with individual neurons conveying very little information. On the other hand, sparse coding could optimise the storage of patterns in associative memory due to the relatively small amount of interference between patterns, and could facilitate the learning of associations due to the fact that localized activity enables biologically plausible local learning, such as Hebbian learning to occur (Rolls and Treves, 1998). In their studies on simultaneously recorded neurons in the macaque IT, Rolls *et al.* (2004) have shown that there is little information in the relative timing of the spiking activity among neurons in IT, and that it is in the firing-rate of the neurons that the bulk of the information is to be found. Their findings also indicate that the neural responses are largely independent with very little redundancy, making decoding by higher processing areas relatively straightforward with a 'dot-product' type of operation between the incoming neural spike-count response vectors and the synapses on the receiving neurons.

Another proposal is that sparse coding is effective for representing the statistical structure in natural scenes. One theory (Barlow, 1972) is that sparseness increases at successive stages of the visual hierarchy, so that at the higher stages, the input is being represented by as few active neurons as possible, without losing any of the information content. However, not all research

substantiates this idea. Baddeley *et al.* (1997), for instance, certainly find that the response distributions of both V1 and IT neurons were sparse, by Olshausen and Field's, (1996) measurement, but find no evidence for increasing sparseness at successive stages of the visual cortical hierarchy between V1 and IT. The authors also suggest that all the firing rates in an ensemble convey useful information, not just those of the most active neurons.

Although redundancy reduction, resulting in dense representation has largely been rejected now as a function of visual processing in cortex (Barlow, 2001) it is still considered to possibly be the principle underlying the coding of signals from the retina up to LGN. This compression is thought to be necessary in order to communicate only the most useful information from the 100 million or so photoreceptors to the approximately 1 million retinal ganglion cells and along the optic nerve to LGN (Olshausen, 2003). However, Barlow argues that the signals from the photoreceptors are much slower than in those in the optic nerve and so the supposed bottleneck may not occur. Either way, V1 represents the information coming up form LGN with about 10^9 neurons (Barlow, 2001).

Olshausen (2003) points out that this suggests an increase in redundancy in cortex, provided the signal capacity of the axons in LGN and V1 is about the same, since the total amount of information cannot be increasing, which implies that the idea is to model the redundancy in visual input rather than reduce it.

Vinje and Gallant (2000) attribute sparse representation in V1, in part, to the interaction of the nCRF, (non-classical receptive field ie the surround) with the CRF (centre), during viewing of natural images. They found that the sparseness of the response of individual V1 neurons is considerably increased when the stimuli span both the CRF and the nCRF, observing enhancing as well as suppressive effects. Both cross- and iso-orientation of the surround stimuli can give rise to either enhancement or suppression depending on the relative contrast between the CRF and nCRF (Yu *et al.*, 2003) and leads also to a sparse representation across the population of neurons in which the neural responses are largely independent with little overlap in neural

selectivity. They also suggest that their results lend weight to the view that V1 neurons might be representing the independent components of natural scenes.

Their work substantiates the findings of Olshausen and Field (1996) whose model optimises sparseness while keeping the information content fixed and thus obtains a set of components, or spatial features, that have similar characteristics to simple cell receptive fields in V1 – localised, orientation selective and band-pass, ie sensitive to different structure at various spatial scales. These components are optimised through the processing of natural image statistics sampled across many thousands of natural image patches.

Another aspect of the representation that Olshausen (2003) stresses is 'overcompleteness' of the set of basis functions or spatial features used to represent an image. Visual processing models often use a 'critically-sampled' representation, where the number of input and output dimensions is equal, so as to achieve linear independence and hence unique representation of incoming patterns. However, V1 appears to use an overcomplete representation of the inputs from LGN, with more output dimensions than input ones (Olshausen and Field, 2005). In a linear system of over-representation the basis functions or neural responses are not linearly independent, and this can lead to ambiguity in the output, however, having an abundance of, say, orientation dimensions or different spatial scales, has the advantage of allowing structure to be represented more precisely. Olshausen and Field argue therefore, that part of the function of the observed non-linear behaviour – lateral inhibition and nCRF/CRF interaction - of V1 neurons, is to reduce the redundancy so that only a few neurons with the most salient selectivity are active in response to a given stimulus.

Hoyer and Hyvarinan (2002) have modelled a different type of non-linearity in the responses of V1 neurons. The model extends the linear sparse coding systems of Olshausen and Field (1996) and others to include a higher layer of neurons that calculate the variances or the squares of the outputs of the simple cell layer and thus display some non-linear characteristics of complex cells such as invariance to position or to reversal of contrast polarity. A further development of the system to obtain higher-level contour-coding neurons, which, it is suggested, could be in V2,

learns a sparse coding of the output from model complex cells, in response to a set of natural image patches, in terms of linear combinations of basis vectors. The authors view feedback from the contour-coding layer during the learning process as top-down inference for reducing ambiguity or noise in the bottom-up representation from the lower layers.

As Lehky *et al* (2005) point out, non-linear transforms such as those operating in the complex cells in Hoyer and Hyvarinen's model do not preserve all the information in the input signal and suggest therefore that there is more than just efficient coding in the presence of noise going on the visual system, especially at the higher levels. Sparseness, as for example Rolls *et al.* (2004) suggest, can increase storage capacity in associative memory. It can also cut down on metabolic demands (Lennie, 2003). Lehky *et al.* speculate that neural representation further up the visual hierarchy may not only emerge 'bottom-up' in response to image statistics, but may also be influenced top-down by requirements for effective association between visual, motor and other mechanisms that are important for survival of the organism.

2.8 Visual perception as inference

It has been hypothesized by Helmholtz (1867) that visual perception relies on a process of "unconscious inference" in order to enable the pattern of retinal stimulation to be meaningfully interpreted. In other words, we rely on assumptions of which we are unaware to 'fill-out' inadequate 2-dimensional optical information to form a 3-dimensional interpretation of the environment. The idea that the overall 'gist' of a scene is perceived explicitly 'at a glance', at a high visual level after rapid feed-forward processing, and that feed-back to lower areas later informs the process of 'filling in the detail', is put forward by Hochstein and Ahissar (2002) in their Reverse Hierarchy Theory.

Kersten *et al.* (2004) suggest that higher visual areas may be representing hypotheses about the contents of a scene. These hypotheses could be used to resolve conflict, due to ambiguities, caused by occlusion, differences in illumination or viewing angle etc, in the representation of image features in the lower levels, such as V1. One approach (Grill-Spector, 2003) is that

feedback to the lower levels with predictions of the likely early-stage activity is compared with the actual response and a signal conveying the error or residue sent back up to the higher levels, possibly the lateral occipital complex (LOC) (which is similar in object recognition function to monkey IT, Riesenhuber and Poggio, 2002), so that the predictions can be modified accordingly. This might imply low levels of activity in areas like V1 when the predictions are a good match for the image measurements (Rao and Ballard, 1999) which would fit with theories that the feedforward and feedback connections in the visual system may be involved in 'explaining away' perceptual ambiguity through the resolution of competition between conflicting hypotheses about the input image (Kersten *et al.*, 2004 and Lee and Mumford, 2003).

There is evidence for this reduced activity in V1 in the results of experiments conducted by Murray *et al*, (2004), using fMRI (functional Magnetic Resonance Imaging). (Functional MRI (fMRI) is a form of brain imaging that shows changes in blood oxygenation during neural activation, enabling researchers to determine the degree of activity in different areas of the brain during certain tasks.) Kersten *et al.* (2004) suggest two possible reasons for the reduction in early visual processing activity. The first is that, as the higher areas explain away the ambiguities in the image, they suppress the lower-level activity altogether in order to save energy, since the metabolic cost of neural spiking is high and energy resources are limited, Lennie, (2003). This would be consistent with Rao and Ballard's (1999) model of low levels of activity when the 'error' between input and prediction is small. Alternatively, Lee and Mumford's work, (Lee and Mumford, 2003) supports the idea that the higher-level areas suppress the activity in the early visual areas that is inconsistent with the higher-level predictions in order to clarify perception.

As well as the reduction of the response in V1, Murray *et al.* (2004) have found a significant increase in the activity in the LOC when whole objects are being perceived through the local grouping of visual features and Lerner *et al.* (2002) have found a gradually increasing fMRI activation across the V1 - LOC hierarchy in response to whole as opposed to scrambled versions of objects. Murray *et al.* suggest that these experimental results lend weight to the view that the

various visual areas maintain multiple 'beliefs' or hypotheses which are communicated and modified through feedforward and feedback processing until an overall high-level perceptual decision is reached, at which point the redundant hypotheses can 'collapse', resulting in a reduction in activity in the lower visual areas. In fact, this is broadly the stance of Lee and Mumford (2003) who explain that the maintenance of multiple possible local solutions to feature values is an approach taken by machine vision systems to prevent sub-optimal decisions being made before global factors can be taken into consideration, in applications such as tracking moving objects against cluttered backgrounds and robot navigation. This technique is known as particle filtering and the basic idea of it is to approximate the full probability distribution on all possible outcomes by the weighted sum of a manageable-sized representative set of 'guesses'. The authors hypothesize a neural implementation of particle filtering and belief propagation in which the pyramidal cells in superficial cortical layers 2 and 3 are responsible for conveying bottom-up signals, while those deep in layer 5 transmit the top-down messages. They suggest that the top-down signals could use the same feed-back mechanism that is postulated as effecting attentional biased competition, discussed in Section 2.9, and further, that the role of attention might be considered within this particle-filtering/belief-propagation framework to be that of 'biasing inference'.

2.9 The feature-binding problem and selective visual attention

2.9.1 Feature-binding

Any model of representation of objects as collections of parts or features requires a means of assembling those parts in an appropriate way. With structural approaches like those of Marr (1982) and Biederman (1987) for instance, parts are bound together by rules of syntax – one part is on-top-of or to-the-left-of another, for example. In a feature-based hierarchy, with its neural implementation, neurons at successive levels are responsive to increasingly complex combinations of features, with ever larger receptive fields covering more of the visual field, allowing greater tolerance to variation in location, view, size etc., to be built in.

However increasing insensitivity to various transformations can lead to the loss of information about the precise spatial configuration of the features constituting an object with the result that objects composed of the same set of features, but differently arranged, cannot be distinguished from one another. This is the binding problem, highlighted by von der Malsburg (1999). To illustrate, there would be no way to differentiate between an object of the form 'triangle on-top-of square' from another comprising 'square on-top-of triangle' and no way of distinguishing different sizes or views of the constituent parts, which could lead to dire consequences for an organism if such distinctions were critical. To overcome this difficulty, von der Malsburg advocates 'dynamic link binding' (von der Malsburg, 1999; Zhu and von der Malsburg, 2004) which entails reorganizing the connections between primary visual cortex and the higher visual areas to eradicate ambiguous connections. Temporal signal correlations are formed at an early stage of processing when the relations among features are still apparent. Different spatial configurations of features give rise to different binding patterns. During the matching process with patterns stored, say, in IT, links between corresponding points are stabilized while links between non-corresponding points are temporarily suppressed. Von der Malsburg points out that temporal synchrony for feature binding is slow and suggests that connector cells that each control the connection between a fixed pair of neurons could be the basis for rapid activation of connectivity patterns. However, he admits it is unclear how such cells could develop in the brain.

On the other hand, Elliffe *et al.*(2002) in their feature-based approach, advocate low-order combinations from a small alphabet of features, containing some spatial information, being built up into ever more complex configurations at each stage as a means of avoiding ambiguity in the representations of different objects.

Another issue for feature hierarchies is the potential occurrence of false binding errors (Elliffe *et al.*, 2002). The difficulty here is being able to detect a target object among multiple other objects containing subsets of features in common with the target and hence collectively containing all the target's features in distributed form. In this case, the problem is how a neuron in one layer can respond to just its preferred spatial configuration of the features. VisNet,

(Elliffe *et al.*, 2002) addresses this problem through lateral inhibition in two ways. First, the various subsets of target features at one level of the hierarchy, say layer N, will be combined in other ways than just those appropriate for forming the target object in layer N+1, and hence, other neurons responsive to these alternative configurations in layer N+1 will tend to inhibit the activity of the target neuron, which helps to enhance the selectivity of individual neurons. Second, the spurious construction of an object through false binding of lower level features tends to involve more features than does the correct conjunction. Lateral inhibition can increase the sparseness of the representation, thus reducing the likelihood of too much activation of lower-level features.

The problem Riesenhuber and Poggio (1999b), address is that neurons in the higher levels ventral visual stream, V4 and IT, would be unable to represent several objects simultaneously, without ambiguity if they simply performed a weighted linear sum of all their inputs. Hence the authors postulate that some sort of non-linear mechanism along the lines of their MAX operator (Riesenhuber and Poggio, 1999a) may modify the activation of certain transform-invariant neurons, so that they only respond to the strongest of their inputs, thus reducing interference from afferents that are activated by non-preferred stimuli of the receiving neuron. A possible MAX function might scan over inputs of the same feature type but under different transformations – size, view etc – and select the most active. One possible way this could be implemented neurally is through making neurons sensitive to the timing of the arrival of signals on their afferents (Rousselet *et al.*, 2004). It is generally the case that the more rapid the firing rate of a sending neuron, the earlier its signal tends to arrive at the receiving neuron (Gawne and Martin, 2002). Hence latency could form part of a MAX operation. Gawne and Martin have found some evidence for non-linear MAX-type operations in some V4 neurons with pairs of stimuli presented simultaneously.

The findings of psychological studies of VanRullen *et al.* (2005), that there is less interference between simultaneously presented familiar 'natural' objects than artificial ones regardless of the degree of separation within the visual field, suggest that there may be two distinct types of feature-binding: hard-wired binding for familiar natural and man-made objects; and binding that

requires attentional mechanisms for less familiar or synthetic objects. The combinatorial explosion problem prevents hierarchical systems from representing, at one level, all the possible combinations of features from the previous stage. The authors suggest that the system may only hard-wire the most relevant objects, possibly determined by practice and experience. Therefore, the possibility that natural objects and scenes are more likely to be coded for than synthetic ones could explain their findings.

2.9.2 Visual selective attention

For animals, such as primates, with highly-developed brains and extensive sensory and motor capabilities, some kind of attentional mechanism is essential for efficiently selecting the most important or interesting information upon which to act, from among potentially overwhelming input to the senses from the environment at any given time. One form of appropriate behaviour might involve directing the gaze towards the source of the most salient signal in order to obtain further information as to its nature before initiating a response (Itti and Koch, 2001). Hermann von Helmholtz (1925) postulated that we do not simply passively receive visual input, but that we can fix our attention at precise points, focusing our eyes on each aspect of an object or of multiple objects, in series. This gave rise to the metaphor of the 'spotlight of attention" illuminating part of the visual field so that the contents can be processed to a higher level of detail to the exclusion of stimuli lying outwith the beam. On the other hand, William James (1890, p6), saw attention as having an involuntary, passive and distributed aspect as well as the voluntary, active and focused form described by Helmholtz. These early ideas led to the classical view of attention, in which there are two distinct phases, an involuntary, rapid, data-driven, parallel processing stage during which salient regions of the input image are identified, followed by a slower, voluntary, largely top-down, serial processing of the individual salient points (Rolls and Deco, 2002, p127 – 128).

Treisman's 'feature integration theory' (FIT) (Treisman and Gelade, 1980) is based on this classical approach. and provides a possible explanation of the results of psychophysical experiments in visual search. The first type of search involves the detection of a target object

that differs from surrounding distractors in a unique feature. In these circumstances, the target 'pops out' of the scene immediately regardless of the number of distractors. In the second type of search, termed 'conjunction search', the target and distractors have one or more features in common, which makes the target harder to find, which is apparent in the time taken to conclude the search – increasing linearly with the number of distractors.

In Treisman's model, the various features – colour, orientation etc – are represented in a set of retinotopically organized feature maps, the contents of which are established during the pre-attentive, bottom-up, parallel processing stage. There is no conscious access to these individual maps except through serial search of the master map. If there is a unique feature distinguishing the target from the distractors, the single map for that feature is activated and the target's location can be read directly from it. However, if the target and distractors have some features in common, no single feature map can provide the location information to guide the spotlight to the target and so a serial search of all the objects is necessary. The model also requires the spotlight of attention for binding the appropriate features together. Outwith the spotlight, it is not clear which features belong together, and this can lead to the formation of illusory conjunctions (Koch, 2004). However there are some problems with the feature integration model. Some researchers have found that conjunction search is not always necessarily serial. For instance, shape and motion can be processed in parallel (McLeod, Driver and Crisp, 1988). Also, Triesman (1988) and Duncan and Humphreys (1992) have found that when the background objects are sufficiently similar, there is little difference in performance in feature and conjunction search. In addition, grouping effects, especially involving coherent motion of items with a common feature can influence search times. Also, FIT makes no use of any user-related, behaviourally relevant information as to the nature of the object or features being sought (Wolfe, 2003).

The 'guided search model' (Wolfe, 1994) addresses these concerns about the division of the search process into parallel, preattentive search and serial attentive search phases. The theory proposes that preattentive processes can guide the deployment of the subsequent serial attentive stage towards salient items (Wolfe, 2003). There are two mechanisms employed in guiding

attention: bottom-up stimulus-driven activation and top-down user-driven activation (Wolfe, 1996). Bottom-up activation is modulated by the degree of difference between an item and its neighbours. Wolfe points out that some features attract bottom-up parallel processing more than others, sudden appearance and novelty for example. However, attention also requires top-down guidance so that information important for the current task can be assimilated without interference from 'pop-out' features in the environment (Wolfe, 1996). The GS2 model can be used to simulate various tasks including feature search, conjunction and serial searches. The input stimulus is filtered through broadly-tuned feature channels – colour, orientation etc, the output from which produces feature maps activated on the basis of bottom-up local differences among objects and top-down behavioural requirements for the location of particular features. An activation map is then formed with peaks corresponding to a weighted sum of the feature activations for each location. Attention mechanisms are then deployed serially, in order of decreasing activation until the target is found, or some criterion is met for terminating the search. Interference due to factors such as similarity of distractors with the target, inhomogeneity of distractors and so on, means that the target does not always give rise to the maximum activation in the activation map, which explains the longer search time previously ascribed to serial search in conjunction tasks.

Most models of visual search use stimuli that are separated from one another and appear against a blank background, but in natural scenes, objects seldom appear conveniently isolated from each other in this way. Wolfe tackles the issue of part-whole structure of objects as well as continuous stimuli and ownership of borders (Wolfe, 1996). His experiments suggest that preattentive processing is sensitive to parts and wholes of object structure. One example he cites is that it is hard to find a house painted half red and half yellow among a set of houses painted red and blue, but it is relatively easy to pick out a red house with yellow windows from among red houses with blue windows. Thus with colour conjunction, search is efficient when the colour of the whole is conjoined with the colour of the part. This implies that the preattentive processing must be sensitive to the objects themselves. Similarly, searches with colour and orientation with continuous and overlapping stimuli have shown that borders tend to

be assigned to the correct objects and the result is unaffected by which object is overlapping which. This ownership of parts, including boundaries, implies that their owners must also be represented preattentively in some form. Since spurious conjunctions of features tend not occur during search, it would seem that features can be assigned to items preattentively. Wolfe (1996) deduces that preattentive processing provides a rough parsing of the visual input into items or objects for subsequent attended processing and proposes an extension of his GS2 model to include preattentive representations of items in a 'preattentive item map'.

The biased competition model of Desimone and Duncan (1995) challenges the need for saliency maps and a spotlight of attention and questions whether the linear increase in search time observed in conjunction search tasks is due to serial processing or to the time required to resolve competition in a parallel search mechanism. It is based on two aspects of the problem of visual attention. The first is the limited capacity to process information, which suggests that giving more attention to one stimulus correspondingly reduces that amount of processing power available to the remaining stimuli. The second is the ability to filter out unwanted information so that only attended objects reach awareness. The authors' hypothesis is that objects appearing in the visual field compete for attention and hence for further processing, with the competition biased in favour of behaviourally relevant input. In their view, attention is an emergent property of these competitive neural mechanisms and is important for the reduction of the ambiguity in the representation of multiple stimuli, especially in the large receptive fields of IT neurons.

Bias operates bottom-up in that a target that differs from its homogeneous neighbours stands out, as do objects that are larger, brighter, or faster-moving etc. This competitive effect may be due, in part, to the fact that the response of a cell presented with its optimal stimulus within its classical receptive field can be suppressed if similar stimuli are also present in the surround. Walker *et al.* (2000) have found that a majority of V1 neurons exhibit suppression of activity in the presence of an optimal grating extended uniformly to cover both the centre and the surround.

Novel stimuli have also been found to command more attention, with the effect tailing off as familiarity increases. In delayed matching-to-sample tasks (Desimone 1996) the responses of

some cells in IT have been found to be suppressed in direct proportion to the degree of similarity between a test sample and one stored in memory. Functional MRI in humans shows that this repetition suppression effect is reflected in a reduction in cortical activation, with fewer neurons involved, when subjects are responding to familiar objects than to stimuli that have not been seen before. This smaller neural activation being associated with better recognition performance suggests that the function of the suppressive mechanism is to enable the remaining active neurons to give a better, sharper representation of the stimulus, an advantage of a sparse representation.

Top-down control of attention in the ventral stream is initiated by the requirements of the task in hand (Desimone and Duncan, 1995). Spatial selection appears to resolve competition between stimuli within the receptive field. In single-cell recording in monkeys, when both target and distractor occur within the receptive field, attention to the target effectively shrinks the receptive field around the target and the responses to the distractor are considerably reduced, whereas attention to the target has no effect on neural response when the distractor lies outside the receptive field (Reynolds, Chelazzi and Desimone, 1999).

With regard to selection on the basis of features, single cell studies of IT neurons have provided evidence that top-down inputs to IT cortex, initiated during cuing, bias competition towards the target. Monkeys attend to and make a saccadic eye movement towards a cued stimulus that subsequently appears with a distractor, both items being located outwith the fovea. The effect of the priming depends on whether or not the target is the cell's preferred stimulus. Cells respond to the target when it appears with the distractor, but then, while the activity of cells selective for the target remains high, competitive interactions cause cells selective for the distractor to be inhibited (Desimone and Duncan, 1995).

A neural model of top-down bias applied to object-based attention is that of Usher and Niebur, (1996). It models delayed-match-to-sample tasks (Desimone and Duncan, 1995). Objects are represented in a sparse distribution, where some neurons are involved in the representation of several objects. Within a 'sensory memory module', neurons sensitive to the same feature are

connected to each other through excitatory connections and the resulting cell assemblies are mutually inhibitory. This is in line with findings in IT, in which the response of a set of neurons responsive to a particular shape is enhanced, while the response of neurons coding for a different shape is suppressed. The model's lateral inhibitory effects are mediated by a common pool of inhibitory neurons. Objects that have no features in common are represented by separate sets of neurons, with the degree of overlap of representation increasing with the similarity of the objects.

For each item in the input layer, for example V1, activation is communicated to the various cell assemblies in IT, the strength of the activation depending on the degree of similarity between the stored object representation and the input item. Weights on the excitatory connections between the cells in each cell assembly are set so that they can generate strong competition among objects through the inhibitory pool. A 'working memory module', presumed to be located in frontal cortex, not explicitly modelled, is considered to have the same architecture as the sensory memory in IT, but with stronger excitation between cells in the same assembly. This enables the response of an activated assembly in working memory to persist in the absence of a stimulus, thus modelling the ability of prefrontal cortex to convey information about an object even when there have been intervening stimuli. Also, a weak excitatory feed-back projection from each working memory cell assembly to its corresponding assembly in the sensory memory is assumed.

In Usher and Niebur's (1996) modelling of a 'delayed-match-to-sample' task, in positive trials – that is, when the target appears in the final display – during the initial presentation of the target, neurons in the target assembly are activated, while those in other assemblies are suppressed, due to competition. The target assembly then communicates with the corresponding assembly in the working memory, which remains active during the delay period and sends a weak 'expectation feedback' signal to the target assembly in the sensory module. When the target and distractor are shown together after the delay, the two corresponding cell assemblies in the sensory memory are highly activated, which in turn activates the inhibitory pool, creating strong competition between the two assemblies. The additional top-down input to the target cell

assembly enables it to win the competition, while the distractor assembly activity is suppressed. In negative trials, in which the target does not appear in the final display, during presentation of the two-shape display, the target assembly is not one of the bottom-up activated assemblies in the sensory memory, so, although it receives the top-down bias from the target assembly in the working memory module, its activity is suppressed. This allows the two distractors to compete for activation, but without the additional attentional bias, neither of them can win.

In agreement with Duncan's late selection theory (Desimone and Duncan, 1995), Usher and Niebur's model shows that the more similar the target and distractors, the slower and less reliable the system becomes in finding the target.

Deco and Zihl's (2001) model extends the work of Usher and Niebur to simulate search times in feature and conjunction searches. Unlike feature integration theory, Deco and Zihl's model does not require a serial search mechanism to explain the linear increase in search time in conjunction searches. It differs from the guided search model in that it does not need serial guidance of attention to relevant areas in a saliency map, the system's competitive mechanisms are implemented during low- as well as higher-level processing of feature information, and all processing is parallel. Their model is related to that of Olshausen *et al.* (1993). However, whereas Olshausen *et al.* route selected input from V1 to higher cortical areas by using control neurons to dynamically modify the strengths of the synapses on intracortical connections, routing of information in Deco and Zihl's system is an emergent property of the parallel competitive processing dynamics.

In order to understand the basics of attentional mechanisms of visual search and selection, researchers have tended to model attention as being separate from representation. However, increasingly, it is being seen that attention and representation are interdependent and that to increase understanding in either area, the effects of each upon the other have to be studied.

To this end, Deco and Rolls (2004) have combined Deco and Zihl's (2001) model of attention with VisNet (Wallis and Rolls, 1997; Rolls and Milward, 2000, for example), to form a system that allows the study of the biasing effects of top-down attentional mechanisms on a pyramidal

feature-based visual hierarchy, with convergent feed-forward connectivity and local competition among neighbouring areas. The goal of the model is to allow analysis of space-based and object-based top-down attention and how the local lateral competition among neurons in the early visual areas becomes increasingly global further up the visual hierarchy.

As with all the models discussed here, Deco and Rolls' system simulates covert attention – ie without eye-movements. It explains the gradual increase in the degree of attentional modulation up through the ventral system found in single cell and fMRI experiments. In V1, lateral inhibition is local, so that objects at a distance do not compete with one another. But in IT, attentional bias has to be applied throughout, so that neurons that respond to a particular stimulus can have their activation enhanced wherever they are in IT. The model also explains the variation in the effective size of the receptive fields of IT neurons in natural, cluttered scenes. It seems that reduction of the receptive field size in a complex scene is effected by both global and local inhibition. Global inhibition in IT causes reduction in firing rates for most stimuli, so that the object at the fovea, where there is a large magnification factor, tends to win. Also local competition in V1, due to the effects of the background in natural scenes, causes an increase in suppressive effects in IT.

This is in agreement with the findings of Kastner and her colleagues (Kastner and Pinsk, 2004) in fMRI studies, that there is an increase in the magnitude of competitive interactions, which is proportional to the increasing receptive field sizes through V1, V2, V4 and TEO, an area between macaque ventral V4 and TE in IT cortex. In V1 and V2 neurons, suppression effects are likely to be due to inhibition in their non-classical receptive fields, whereas in V4 and IT, there are more likely to be multiple stimuli competing for representation within the receptive field. In addition, spatial attention enhances the response to stimuli in that area, by counteracting suppressive influences from neighbouring stimuli competing for limited receptive field resources, thus eliminating unwanted distractor information. Stimuli lying outwith the receptive field have relatively smaller suppressive effects on those within the receptive field, so this supports the idea that when attention is directed to a particular stimulus among multiple stimuli, the receptive field may shrink around the target, effectively excluding the unattended

stimuli from the receptive field. These suppressive mechanisms are found to be strongest within the receptive field, becoming more moderate outwith the receptive field, steadily decreasing with increasing distance from the focus of attention, thus filtering out much of the visual input.

2.10 Conclusions

The literature reviewed in this chapter has posited important theories in several areas as to how biological vision tackles the difficult task of object recognition. Many of these are significant for machine vision research:

- Multilevel representation, with consideration to the type of stimuli or 'features' to which neurons at various levels are sensitive, including shape, colour and texture information
- Increasing complexity at successively higher levels of representation, with associated enhancement of tolerance to variability of input stimuli, in terms of attributes such as size, orientation, location, leading to better generalization ability
- Denseness/sparseness of representation and how attention mechanisms enable the visual system to select relevant features or objects and suppress responses to irrelevant ones, thus preventing an overload of information by controlling the dimensionality of the representation
- Local connectivity at lower processing levels reducing combinatorial problems at higher levels
- Communication between representation levels including feedback from higher levels helping the resolution of ambiguity at lower levels
- Being able to adapt the current representation to learn new things without having to relearn the current repertoire in the process of including new stimuli and in addition only requiring minimal exposure to a new stimulus

Chapter 3 now looks at current machine vision research in the area of object recognition and at how the important aspects of biological vision listed above are informing the various approaches being explored.

Chapter 3: Engineered Machine Vision Systems

3.1 Introduction and overview of machine vision systems

Many different types of visual problem are addressed by machine vision systems – biometric tasks such as iris recognition, face recognition, finger-print recognition. Also tasks such as recognition of handwriting – handwritten characters and cursive script, detecting pedestrians, cars, postcode recognition, image retrieval, identifying people by their gait, medical imaging, analysis of radar images, discrimination of multiple different categories of object, fine discrimination of objects within the same category – faces, facial expressions, within species flora or fauna discrimination.

This chapter examines various approaches to constructing robust machine vision systems, in the context of the major areas of biological vision research identified in Chapter 2 as being significant for artificial systems.

Machine vision systems are broadly comprised of four main components:
1) A pre-processing stage, during which the data from the input scene is converted into a suitable form of signal from which the system can learn the required visual task. There are many possibilities for the type of information extracted and how it is represented.

2) A learning stage, during which a classifier or set of classifiers is trained on the representation prepared in the pre-processing stage. Many different types of classifiers are used. Generally, a specific type of classifier is selected as most suitable for the task in hand. A large number of classifiers are designed to be used in schemes that learn from examples – neural networks, support vector machines – and the learning process can be 'supervised', meaning that the system has a 'teacher' in the sense that the exemplars are provided along with the labels for each class, and 'unsupervised', where no labels are provided, and the system must discover the classes for itself. Usually the number of classes is specified in advance – SOM, probabilistic

classifiers, clustering techniques such as *K*-means, and semi-supervised learning where some labelled data is provided and is augmented with unlabelled examples which are generally more readily available.

3) Classification is then generally carried out by processing a new example in the same way as the training data and passing it to the classifier, usually in the form of a vector of numerical values. The classifier, in turn, outputs a numerical value or set of values in response to the input.

4) A post-processing stage, during which the output from the classification stage is interpreted so that the test item can be assigned to the class to which the system has found it to be most similar.

The choice of representation and classifier can be task dependent. However, it is the aspiration of machine vision research to design all-purpose systems that can detect and classify large numbers of different categories of object under a wide variety of viewing conditions – changes in illumination, location, size, orientation, rotation in depth, background clutter, partial occlusion, noisy images, distortions. The representations sought are those that are invariant under such image transformations. In addition, as is the case with the human visual system, it is increasingly considered that artificial systems should be adaptable to new categories without needing to be redesigned from scratch. Another important aspect of human vision is the ability to learn from a single or just a few examples of the new object (Rolls and Deco, 2002, p120). Often, for machine vision systems, there may not be much data available for training, or there may not be time for an online system to train on many examples, so 'one- shot' learning or learning from a few examples is an important area of research.

The rest of Chapter 3 is arranged as follows: Section 3.2 looks at the overall approach to and aims of extracting information from digital images in the formation of a useful representation for various visual tasks. In Section 3.3, different types of image information or 'features' are discussed, including global and local information, contour-based and texture-based features, biologically-based features and how 'informative' individual features should be.

Section 3.4 investigates the problem of managing potentially large amounts of information using techniques of feature selection.

Systems that model the distribution of data within object classes are compared with those that emphasize the differences between object classes in Section 3.5. Section 3.6 looks at how new classes of object can be learned from a few examples with the help of knowledge about the familiar classes. In Section 3.7, various techniques for segmenting images to aid detection of objects or regions of interest are discussed.

Section 3.8 examines different architectures in object recognition systems, ranging from single-level to multi-level biologically based systems and the arguments for their use.

Conclusions are drawn in Section 3.9, where the achievements of current research are acknowledged and areas in which an alternative approach might point to a possible way towards greater autonomy and adaptability of machine vision systems are identified, leading to the formulation of the research questions of the thesis. These questions are then stated at the end of the chapter.

3.2 Overview of Representation in object detection and recognition

The usual input to a machine vision system is a digital image that has been captured by a camera, satellite, radar, medical imaging scanner and so on. A digital image contains a great deal of information, in the form of individual pixel intensities and colour. This raw data is usually considered to be too much input for a machine vision system to process effectively.

Assuming a set of training images representative of several classes or categories of object, the raw pixel data has the potential to provide information about the inter-class differences and intra-class similarities. However, in real-world problems there can be considerable differences among instances of the same class, and using the pixels directly causes this information to be included as well, which might be useful if fine detail is required, for example, in face

recognition, but otherwise can lead to the representation being 'overfitted' to individual examples with resulting poor generalization. Therefore an efficient way of reducing the quantity of input to manageable proportions while maximizing information about the inter-class variability and intra-class similarity and minimizing any unwanted input about intra-class differences is required. This process is generally referred to as feature extraction and is often performed as a pre-processing stage.

3.2.1 Image preprocessing

Before feature extraction, it can be useful to filter the input image in some way to reduce the effects of noise on the detection of relevant image structure such as edges. The technique, known as 'smoothing', works on the principle that pixels in a neighbourhood should look similar to one another, and adjusts each individual pixel value in the image by setting it to a weighted average of its neighbours. A commonly used smoothing filter is a symmetric Gaussian kernel the weights of which are large at the centre and decay rapidly in the surround, so that the immediate neighbours of a pixel have the greatest influence on its revised value, (Forsyth and Ponce, 2003, p136). Kpalma and Ronsin (2006) apply a Gaussian filter at successively decreasing bandwidths to smooth global contours.

3.2.2 Transformation invariant representation

Bishop (2006) describes four broad approaches to achieving the same output response from a classifier despite various transformations of the input variables.

1. A sufficiently large training set might include exemplars at multiple locations, scales and orientations, otherwise it could be augmented with suitably transformed versions of the original training examples. For instance, for translation invariance, several copies of each training item could be made, with the object shifted to a different position in each. An example of this approach is Opelt *et al.* (2006), which addresses variations in scale and rotation of object parts by using scaled and rotated versions of the features in the codebook. Maree *et al.* (2005) achieve scale invariance by rescaling extracted subwindows to a fixed size, while Belongie *et al.* (2002) normalize the distance measure between pairs of points using the mean distance between

all the point pairs in the shape, in the context of shape-matching. In Fergus *et al.* (2003), the variability within object classes is represented by Gaussian probability density functions that model appearance, scale of the features and relative scale of the object parts as well as occlusion and relative positions of object parts. Shotton *et al.* (2008) represent objects by parts in the form of contour fragments arranged around a central point. Each object in a training image is enclosed in a bounding box, the centre of which is taken as the object centroid. The object scale is taken to be the area of the bounding box, which is then normalized to 1. The scales then used for detecting objects during sliding window classification are based on the range of scales found in the training data.

Kpalma and Ronsin (2006) use the idea of scaling in two senses. The first relates to a smoothed version of an object contour, which having shrunk as a consequence of the smoothing process, is then 'stretched', by means of a 'gain-control' function, into a convex curve that intersects the original unsmoothed curve at a set of points. The scale of the smoothed contour is controlled by the size of the Gaussian smoothing function width, σ. As this increases, the smoothed contour size increases and the set of intersection points with the original contour decreases. For each value of σ there is an intersection pattern that can be used to characterize the input shape for recognition purposes. The second sense of scaling relates to the fact that the derived contour-intersection representation is scale invariant to a wide range of scales because the map of the intersection points does not change significantly over a range of scaling factors.

2. A regularization term can be added to the error or cost function so that there is a penalty for a change in the system output when the input has been transformed.

3. The required invariances can be built into the features themselves during the pre-processing stage. Some of the features discussed in the next section are invariant to several transformations.

4. Invariance can be achieved through the architecture of the classifier. The convolutional neural network of LeCun *et al.* (1999) achieves an element of translation, scale and distortion invariance. A large part of the architecture is comprised of layers containing several feature

maps derived from the input image. Local connection of processing units within small neighbourhoods, and weight-sharing across each feature map causes the same feature extraction process to be conducted at all locations in the image. Subsampling of the feature maps from a layer then reduces the resolution of the outputs, thus increasing the tolerance of the system to distortions and shifts in position. Riesenhuber and Poggio (1999) and Serre *et al.* (2005) also employ system architecture to achieve translation and scale invariance, increasing generalization at successive levels of representation.

3.3 Feature extraction

Despite the considerations mentioned above, some systems do make direct use of raw pixel data to learn a shape-based representation. Keysers *et al.* (2004) apply graph-matching to the task of hand-written digit recognition based on finding best-matching corresponding pixels in a pair of images, while Pham and Smeulders (2006) use pixel matching for face detection. LeCun *et al.* (1999) use the pixel intensities from images as direct input to a type of multi-level neural network system called a convolutional network, that through a process of successive levels of local filtering alternating with subsampling, gradually learns increasingly complex features which are not designed by the user. This involves a great deal of processing capacity and the ability to cope with the considerable variability of the input data. Some systems use simple features like individual pixel intensities or wavelets and overcome the problem of separating the different classes by mapping the representation to a much higher dimensional space in which the classes are more separable using a support vector machine.

Many systems use the extracted features directly (Mel, 1997; Ullman and Sali, 2000) while others employ a parts-based representation and then encode each part as a set of features (Fei-Fei *et al.*, 2007; Fergus *et al.*, 2003).

3.3.1 Feature types

Features or descriptors are useful if they can capture intra-class characteristics and inter-class differences and are invariant to the kinds of image transformations and distortions described above. Different types of features have different invariance attributes.

Features can be global measurements, such as Fourier transforms, as used by Lai *et al.* (2001) for face recognition. A Fourier transform converts an image in the spatial domain to an image in the frequency domain to provide information about what proportions of the image signal are at which frequencies. It is translation invariant, but not scale or rotation invariant. However, Lai *et al.* (2001) derive a rotation and scale invariant version of the transform. This is then applied to low-resolution images obtained through wavelet decomposition, to capture the invariant facial features under these transformations.

A set of wavelet transforms allows images to be represented at different resolutions and scales. For instance, in Lai *et al.* (2001), using the low-frequency wavelet components enhances the global face description and eliminates the higher-frequency detail of differences in facial expression. In pedestrian detection, Oren *et al.* (1997) use wavelet templates at a different resolutions and orientations to encode various types of structural information.

Linear projection techniques such as PCA (Principal Components Analysis) and LDA (Linear Discriminant Analysis) are related approaches to representing data in terms of linear combinations of variables.

Global features provide a compact single vector representation in a high-dimensional space, but they are not tolerant to occlusion and background clutter, and can only be used for images containing a single object, or in conjunction with image segmentation.

However, in answer to this, Oliva and Torralba (2006) build a global representation of the image scene, that does not require segmentation, and is not adversely affected by clutter – in fact, it models clutter as part of the scene description. The representation is based on the pooled outputs of local feature detectors, responsive to oriented edges and textures combined to form a

kind of 'global receptive field', from which the overall gist of the scene can be inferred, in terms of spatial layout properties such as naturalness, open-ness, expansion – based on descriptions that are meaningful to human observers. This global feature is then used in parallel with local representation to help direct the search for objects in cluttered real-world scenes, acting as 'global contextual priming'.

Pham and Smeulders (2006) employ a global approach to pixel-matching in which stable long-distance dependencies between pixel values are learned in face and horse images, using a Bayesian model.

Edge detection is a very important aspect of feature extraction and as discussed in Chapter 2, Section 2.3 of the thesis, detection of oriented edges is a function of low-level biological vision (Hubel, 1995). It is also in line with Marr's computational model of vision in which the first stage of processing a scene is the detection of intensity changes at different scales and directions in deriving what is termed the 'raw primal sketch' (Marr, 1982). Edge detection can be applied in the derivation of both global and local representations.

In image analysis, practically any asymmetric filter or mask can respond to sharp changes in greyscale in a local region. However, detectors that are designed to find edges at a specific orientation and scale are likely to be more efficient.

The Sobel operator is a simple 'discrete differentiation operator' used in edge detection algorithms. It computes the intensity gradient at each image point and indicates the largest intensity change and the approximate direction of that change. It is comprised of two 3x3 kernels, one for detecting horizontal changes and the other for vertical changes. The image is convolved with each kernel in turn and the gradient and direction at each point is estimated by the magnitude of the outputs of the two convolutions and the angle of its orientation. The approximation of the direction is fairly rough as only the immediate neighbours of an image point are involved in the calculation and only integer values are used in the kernels (Fisher *et al.*, 2010).

Probably the best-known, and still much-used, edge detector is that of John Canny (1986) which is gradient-based and was devised to improve on the accuracy of existing detectors at that time. It tackles three important issues in reliable edge detection. The detector should find as many of the true edges as possible, without marking non-edges, it should locate the edges as accurately as possible and it should only indicate a single representative for a given edge.

There are three stages in the detection process. First, the image is convolved with a Gaussian filter to reduce noise through smoothing. The second stage is to detect edges, computing the magnitude of the gradient and quantizing the direction of the response to a horizontal, vertical or diagonal orientation. This stage also applies non-maximum suppression, in which a pixel is only considered to be part of an edge if its intensity gradient in the direction across the edge is greater than it is in either direction along the edge, otherwise it is rejected. The final stage eliminates multiple detections of an edge in a process of edge tracing and hysteresis thresholding. Making use of the directional information obtained during detection, an edge is traced through the image. To start a trace, the intensity gradient at a point must exceed the higher of two thresholds, then while continuing the trace, the lower threshold is applied to determine whether fainter responses should be considered as part of the same edge. Edge information is also extracted at a number of different scales. The problem is that different sizes of operators mark edges at slightly different locations, and so to overcome this, an approach termed 'feature synthesis' is employed. This uses the responses of the smaller filters to predict those of the largest operator if the edges detected by the small operators were the only ones in the image. The actual response of the largest operator is then compared to the synthesized response, and only if the actual output exceeds the strength of the synthesized output by a significant amount are any new edges marked (Canny, 1986).

The Canny edge detector is a first-order detector since it uses the intensity gradient. Another approach is to compute the rate of change of the intensity gradient and thus detect edge points as local maxima in the gradient. An early example of this is the Marr-Hildreth operator that applies a Laplacian operator to a Gaussian-smoothed image (Fisher *et al.*, 2010). Methods

based on differential geometry overcome problems of false detections and can detect edges to sub-pixel accuracy.

Edge detection has a high computational cost and forming meaningful constructs from the potentially large number of edges in an image requires further processing to extract the required shape information.

One approach to this was devised by P. V. C. Hough in 1962. The original Hough transform was designed to detect straight lines in images. The idea is that the problem of finding sets of collinear points in the image plane is transformed into the task of finding concurrent lines in a parameter space. Due to the particular parametrization Hough employed there were some problems with the implementation, so in 1971 Duda and Hart devised an alternative parametrization that described a potential line through a given point in the image plane in terms of its polar coordinates. This enabled each point (x, y) to be represented by a curve in parameter space representing all possible orientations of line through (x, y). Points of concurrence of curves in the parameter space indicate subsets of collinear points in the image plane. Searching for all possible lines is prohibitive, and so the search space is quantized into a two-dimensional array of accumulator 'bins' that count the number of curves that pass through them. Each time a bin count is incremented, this is considered as a vote for the shape to be detected. The accumulator array cells are then inspected for high counts that are strong indicators of the presence of lines. A total of k curves passing through a cell indicates that there are k points in the image that are roughly collinear (Duda and Hart, 1971).

Ballard (1980) describes the parametrization for circles and ellipses and parabolas and introduces the generalized Hough transform for detecting arbitrary shapes. This is a non-parametric approach that uses a look-up table to define the relationship between the boundary points and the Hough parameters in relation to an arbitrary reference point for the shape. The table is derived from a prototype of the shape to be detected and the information is stored as distance and direction pairs associated with the known orientation angles of the boundary. The Hough space indicates possible locations of the shape in the image. Differences in scale can be

accommodated through scaling of the look-up table vector entries by the required scale-factor, and rotations of the shape can be effected by rotating the vectors in the table by the required angle (Ballard, 1980).

The Hough transform has some limitations. Setting an appropriate size and shape of the accumulator bins is crucial for capturing a high number of votes for the potential detections. Similarly, having too many parameters in the Hough voting space causes the distribution of votes to be sparse. In addition, the method is very sensitive to noisy edges but it can cope with missing boundary sections.

Opelt *et al.* (2006), in an application to detect and discriminate horses and cows, use the Canny edge detector to find linked edges as candidate object boundary fragments in the training images. The generalized Hough transform is then applied to enable weak detectors representing small subsets of boundary fragments extracted from the edge images to vote for the centroid of the shape their boundary features potentially belong to and contribute to the decision of the overall strong detector using a probabilistic scoring approach.

Shotton *et al.* (2008), also use the Canny edge detector to generate an edge map from which contour fragments are derived using edge gradient information. However, the object detection is achieved by a sliding window approach in location and scale space and incorporating mean shift mode detection.

Murphy *et al.* (2003), in a combined global and local representation with the same motivation as that in Oliva and Torralba (2006) namely, to use information about the whole scene to help resolve ambiguity at the local level, employ the same set of feature extractors to encode both the global and the local representations. The local components are patches which are smoothed and then oriented edge and corners and long edge filters are applied. The global feature vector is obtained by treating the whole image is if it were a patch, extracting the features and then selecting a subset of the best features to reduce the resulting vector to a manageable size.

Once edge and contour information has been extracted from an image, it can be used in the form of a template, either globally or locally, for comparison with images or subregions. Templates can also be derived directly from greyscale or colour images.

The template is moved across the image like a 'sliding window' and the difference measure is calculated at each location. The location for which the difference is smallest is taken as the required match. The approach is translation invariant but computationally expensive. The process can be made more efficient by reducing the number of locations at which matching is carried out. One way to achieve this is through the use of an interest operator to detect locations that are likely to contain an instance of the required object or part. Various difference measures are used, such as Euclidean distance, or normalized cross-correlation as in Brunelli and Poggio, (1993).

Templates are often applied at a variety of scales and orientations and in order to reduce the amount of computation, an image pyramid is generated. This type of representation is achieved through repeated filtering and subsampling of the original image to produce a set of images of increasingly low resolution. Lower-resolution templates, at a variety of orientations, are then applied to the correspondingly low-resolution, smaller images to find possible locations to begin searching for a higher-resolution match in the larger-scale images (Forsyth and Ponce, p159).

Shotton *et al.* (2008) use contour fragments as templates for matching parts of horses in a star constellation model. An edge map and a template are compared using the distance from the edge pixels in the template to their nearest edge pixel in the edge map and also comparing their corresponding orientations. The contour fragment templates are extracted at various scales and so multi-scale matching is achieved by normalizing the scale of the templates for comparison with the unscaled original edge-map.

The generative, probabilistic model of Fergus *et al.* (2003) learns a set of representative image patches as templates for parts of objects of several different class types, including faces and motorbikes. The saliency detector of Kadir and Brady (2001) is applied to the task of finding regions of interest at various locations and scales. These regions are then adjusted to a fixed

scale. Brunelli and Poggio (1993) investigate the use of eyes, nose, mouth and whole face templates for face recognition and also compare templates with a feature-based approach using geometrical face measurements and finds the local templates more reliable.

Ullman and Sali (2000) extract overlapping class-specific image fragments of varying complexity using training examples of faces and cars in different views. Fragments of intermediate complexity are then selected using mutual information, as described in Section 3.3 below. The fragments are collected into equivalence classes of different views of the same region or part of an object under a variety of transformations and viewing conditions. Detection is conducted at several scales with each pixel in a fragment view being matched to the closest pixel in a small neighbourhood around a corresponding test location. Comparison within a region is on the basis of shape similarity, estimated by a weighted sum of the displacements of pixels with the same ordinal order of greyscale value between that region and the fragment. The absolute orientations and gradient differences are also used.

For tackling the problem of object identification, such as recognizing a particular face, under different viewing conditions, Ullman and Bart (2004) derive sets of 'extended' features from moving images to provide the required invariances. These fragments are larger and more complex than the fragments of Ullman and Sali. In this work, image patches are compared with fragments features using normalized cross-correlation of greyscales. As in the previous work, multiple template fragments of parts under different transformations are collected into equivalence classes for invariant detection, but now the correspondence between face regions across different views is obtained using a motion tracking algorithm. The extended fragments are compared with invariant fragments. As with the extended fragments, the invariant fragments are selected using mutual information, but a single feature is now required to generalize to frontal and side views. Informative invariant fragments occur less frequently, are differently distributed across the face images and provide less mutual information than the extended features, which are shown to perform better for face recognition under different views. The suitability of different features for different tasks as well as the obtaining an optimal subset from among a large set of extracted features is discussed in Section 3.3 of the thesis.

Epshtein and Ullman (2005) have extended the fragment-based representation to a feature hierarchy. The features are extracted top-down, so that large and very class-specific fragments that can rarely be found in images that do not contain the object are derived first, then the requirement is that sub-features should appear often as part of the larger 'parent' feature, but only infrequently elsewhere. At each successive level, distinctive sub-features are sought and the process terminates when further decomposition into simpler features is no longer providing useful additional information. Classification of features is conducted bottom-up, using an HMAX model similar to that of Riesenhuber and Poggio (1999) with alternating levels performing weighted-sum and 'max' operations.

This feature hierarchy model is further developed (Epshtein and Ullman, 2007) as a 'semantic' hierarchy, which is a 'probabilistic graphical' model in which the suitability of a new feature or part is considered in the context of the rest of the parts in the hierarchy. The feature selection approaches in these hierarchical models are discussed in Section 3.3 of the thesis.

SIFT (Lowe, 1999) is short for "scale invariant feature transform". The features are histogram-based, and are translation, scale, rotation invariant and tolerant of some variation in illumination and of affine projection and because they are local and generated in large numbers they also allow for partial occlusion and tolerance to clutter. They are considered by Lowe as being akin to features of intermediate complexity represented in the responses of neurons in primate inferior temporal cortex to very specific stimuli such as dark five-sided star shapes.

SIFT keypoints are extracted from regions of high variation at different scales within a set of reference images, and stored. Each point is defined at a particular scale and orientation. Tolerance to local geometric distortions at different scales is achieved by allowing orientation to vary within a small region surrounding a keypoint at different levels of a scale pyramid. For matching features in new images with the stored keypoint vectors, a nearest neighbour matching approach is used in a search space reduced by a 'best bin first' search method. The Hough transform is then applied to cluster features into subsets that each indicate a particular pose of a single object. The best-matching candidate clusters are determined through a least squares

approach to finding the cluster of points that requires the minimum affine transformations to match the image features to the hypothesized model.

SIFT features are useful for finding matching regions in images and are used in a number of applications including 2-dimensional object recognition, determining the location of a robot in an unknown environment, recognition of human actions and analysis of 3-dimensional MRI (magnetic resonance imaging) brain scan images (Wikipedia, 2010, "Shift invariant feature transform"). However, one or two potential limitations have been pointed out. Belongie *et al.* (2002) point out that keypoints are not suitable for all object types, for example objects with very smooth contours such as circles. The approach adopted to shape matching in this work is therefore to select contour points that do not necessarily correspond to points of maximum curvature. Serre *et al.* (2005) note that SIFT features are useful for detecting familiar objects under new transformations, but deduces from experimental results of comparison of the performance of their higher-level 'C2' features with SIFT on several object categories, that SIFT may be too invariant to cope with generic object recognition tasks.

A different type of intermediate-level feature that does not involve histograms of edge orientations as does SIFT, but is instead based on Gestalt principles of grouping, is introduced in Bileschi and Wolf (2007). Four features are derived, based on the principles of continuity, symmetry, closure and repetition, to be used alongside the C1 features of Riesenhuber and Poggio and the Histogram of Oriented Gradients features of Dalal and Triggs, described below, with the aim of enhancing performance.

Fundamental to the Gestalt principles of form perception is the idea that, given that certain 'rules' are obeyed among configurations of elements, specific grouping patterns tend to emerge.

The principle of continuity is based on the idea that, for example, lines that follow a smooth continuous path are preferred to those that make a sharp change in direction, Figure 3.1.

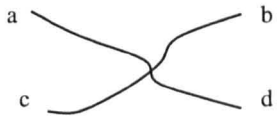


Figure 3.1: The Gestalt principle of continuity
The curves from a to d and from b to c are more likely to be seen as lines than those connecting a to c and b to d, from www.artinarch.com.

The principle of symmetry tends to group together pairs of elements where one is a mirror image of the other about an axis of symmetry, Figure 3.2.

Figure 3.2: Gestalt principle of symmetry
The shapes tend to be grouped as pairs of closed brackets rather than being grouped by proximity, Soegaard, 2010.

The principle of closure tends to make us fill in the 'gaps' between separate objects or parts that when taken together as one, form a familiar shape (Soegaard, 2010, Figure 3.3).

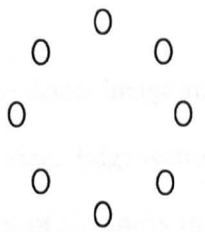


Figure 3.3: Principle of closure
'Gaps' are filled between separate objects or parts to form a familiar shape.

The principle of similarity or repetition causes similar shapes to be grouped together, Figure 3.4.

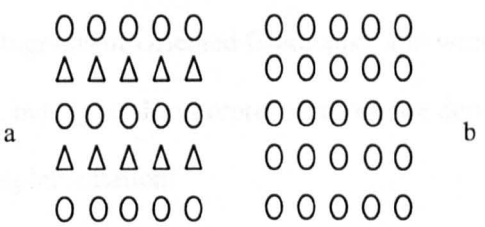


Figure 3.4: Principle of repetition
Similar shapes are grouped together. 'a' tends to be seen as alternating rows of circles and triangles, whereas 'b' is seen as a single square block.

The continuity-based descriptors are obtained using the mathematical morphology operations of erosion and dilation followed by reduction on image resolution in an iterative process to

produce a global feature vector , the elements of which are responsive to contours of a given length and orientation at a particular spatial location (Bileschi and Wolf, 2007).

The closure-based descriptors are designed to represent roughly circular closed shapes in the image. The approach to detecting convex shapes is similar to that of the generalized Hough transform in that the orientation information associated with each edge element is used to restrict the voting in the Hough circle parameter space. However, Bileschi and Wolf add a maximal suppression stage, in which the votes from the highest contributing orientation at each point in the parameter space are removed, which has the effect of reducing detections of straight lines when only closed forms are intended to be represented.

The symmetry feature vector is comprised of symmetry scores calculated at various image locations and scales. An image patch is compared with reflected patches extracted from the opposite side of a potential vertical axis of symmetry and scores associated with the same axis are summed.

The repetition feature is designed to detect image similarity within a neighbourhood, based on comparison of patches of a similar size. Edge responses of similar strength at a given orientation are sought. A consensus of similarity in a local region is computed by pooling the responses and determining the extent of the translation within that region for which the similarity response is maximum.

These intermediate-level features are built using the histogram-of-orientations-based approach of SIFT and HoG (Histograms of Oriented Gradients), and with a similar maximum operation to that of the C1 features, but higher-level representations are derived through the application of small changes in the implementation.

Histograms of Oriented Gradients (HoG) features (Dalal and Triggs, 2005) were devised for pedestrian detection. They are similar to edge orientation histograms (Freeman and Roth, 1995), SIFT and shape contexts (Belongie *et al.*, 2002), but the representation is derived from a dense tiling of the image with a uniformly-spaced grid of 'cells' rather than through a set of

relatively sparsely distributed 'interest points', and it uses contrast-normalization over larger, overlapping subregions or 'blocks' to enhance performance.

Edge orientation histograms were applied by Freeman and Roth to the task of static hand gesture recognition (Freeman and Roth, 1995). The features are computable in real time, translation invariant and tolerant of illumination changes. The orientation is determined at each pixel and a histogram of quantized orientation occurrences is computed. The raw histogram is then blurred to allow increased generalization ability.

Dalal and Triggs (2005) use this approach to computing gradients and incorporates a spatial element in that the orientation histograms are computed for each local cell of pixels. As the authors point out, the strength of HoG features for pedestrian detection lies in sampling over lots of different orientations while allowing spatial flexibility through a coarse binning of locations and using contrast normalization over relatively large overlapping regions to provide a representation that is tolerant of illumination changes and variation in body poses, as long as the main body orientation is upright. The authors have since adapted the technique for detection of animals and vehicles in static images and detection of humans in video sequences.

The 'shape context' descriptor of Belongie *et al.* (2002), mentioned above, provides a rich description of an object's shape by representing each point, p_i, of a finite subset of n contour points in terms of its distance and direction to the other n – 1 points. To reduce the amount of detail captured by this representation, a coarse histogram of the distribution of the locations of the other points to the point is computed on a uniform log-polar grid. This allows better generalization among shapes of the same category and the log-polar representation has the effect of making the histograms of points in nearby locations on shapes being compared, more similar than those of points further away from each other. The features are translation and scale invariant and can be made totally rotation invariant, dependent upon the application. They are also robust with respect to small local variations in the shape, noise and outliers. They are applied to the tasks of retrieving the similarity of the silhouettes of shapes, 3-dimensional object recognition under different views and trademark retrieval.

Similarity between points on a pair of shapes under comparison is computed as a cost function of matching the shape context histograms of the two points. Bipartite graph matching is then applied to minimize the total cost of matching all pairs of points.

The 'C1' features of Riesenhuber and Poggio, (1999), were originally derived from the theory of Hubel and Wiesel (Hubel, 1995), that a complex cell's translationally invariant response could be obtained from the smaller phase-dependent receptive fields of neighbouring simple cells feeding into it. These C1 'pooling units' employ a non-linear MAX-type operation, to achieve a degree of transformation invariance, by scanning over the afferent outputs of template-matching S1 units, that vary over the transformation under consideration, such as position or scale, to find the largest response, indicating the best matching afferent. The idea is that, especially at lower visual levels, the afferents to a pooling cell, that are perhaps processing information from different spatial locations might be responding to different objects or different parts of a single object, and so the standard approach of pooling by summing the afferents could lead to these responses from the different stimuli being 'mixed up'. The same process of template matching and MAX-pooling is repeated in the model to produce complex composite cells, referred to as C2 units. These respond to combined stimuli with yet greater tolerance to transformations. This hierarchical representation is discussed further in Section 3.7.5 of the thesis.

The model is extended by Serre *et al.* (2005), in that, instead of a fixed dictionary of features devised by hand and applied to a set of segmented objects, a vocabulary of features is learned from images and used to classify real-world images.

Bileschi and Wolf (2007) liken C1 features to histogram bins, with each cell computing a function of the incoming filter responses representing the different orientations, but only taking the maximum response rather than forming a sum of responses.

The type of features extracted generally depends on the application – what degree of tolerance of image transformations is needed, whether the task is binary classification or a multiple class

discrimination problem, whether good generalization or the ability to make fine distinctions between different instances of the same class is required, what the system architecture is to be and what type of classifier is preferred. Closely related to these considerations is the problem of deciding on the degree of complexity, or how informative the features should be.

3.3.2 Generic versus class-specific features

Very simple features, in the extreme, individual pixels, provide only a little information, and, as suggested above, are quite likely to encode noise if used directly. At the other end of the scale are the 'informative' features of Vidal-Naquet and Ullman (2003), image patches that maximize the amount of mutual information between patch and image of the class to be learned, making them 'class-specific'. Somewhere in between are, for example, the Haar wavelet features used by Oren *et al.* (1997) and the 'rectangle' features of Viola and Jones (2001).

The generic approach uses a vocabulary of simple features that tend to appear relatively frequently in digital images and can therefore be applied to every class, with each class being characterized by different combinations of these features. Marr's computational theory of the visual system, based on the findings of Hubel and Wiesel, suggests that the lower levels of the ventral system compute generic information about oriented edges, corners, blobs, as in the *Raw Primal Sketch* (Marr, 1982, p71).

Vidal-Naquet and Ullman (2003) argue the that use of 'informative' features of intermediate size means that only a simple linear classifier is required instead of the more complex models needed to combine the simpler, generic features into more meaningful constructs.

Torralba *et al.* (2007) find that when a classifier is trained to discriminate a single class from all the other classes, one-v-all classification, that informative, class-specific features work best, but when a classifier is trained on several object classes, to keep the number of features required under control, it is better to use features that are more generic that can be shared among similar classes.

In Maree *et al.* (2005) the approach is to randomly extract large numbers of patches or 'sub-windows' as features, encoding them in terms of their raw colour pixel values. These sub-windows are labelled with the class of the image of origin, and are thus class-specific, but there is no refinement of the initial selection on the basis of any 'informativeness' measure, such as that of Vidal-Naquet and Ullman (2003). These features are used to train an ensemble of decision trees for the recognition of multiple categories of object.

One problem with the more informative, 'class-specific' features is that a separate set of features is required for each class, which involves a large amount of processing, since the approach is to generate, for each class, a great many of these features, many of which are redundant and therefore need to be eliminated from the final feature set.

However, one difficulty with the generic features is knowing which to combine to form the increasingly complex and more abstract structures that occur at higher levels of representation. A way round this problem is simply to take a great many measurements of different types – edges, greyscale gradients, colour, texture, curvature, angles between edges forming corners – as in Mel (1997) and hope that there will be a subset of features that can characterize each object.

Serre *et al.* (2006) apply a redundant 'universal dictionary of features' to multi-class classification with the CalTech101 database. It is found that, while the class-specific features derived from the positive training data perform better when there are sufficient training examples, the universal feature set performs better when only relatively few training examples are available and fewer features are needed.

3.3.3 Shared features

In between fully generic systems and class-specific ones is a set of models that employs both approaches. While some features they employ are class-specific, others are shared among subsets of classes and the system learns which features are the best for sharing among which classes.

Storing a unique set of features for each class of object to be learned could soon lead to an unwieldy system as the repertoire of object categories increased. Torralba *et al.* (2007) tackle the problem of detecting and recognizing a wide variety of object classes and different views in cluttered scenes with a system that keeps the size of the feature set manageable by sharing features among class representations. The idea is to learn which subsets of classes should share particular features in order to reduce the classification error. First a large set of fragments of various sizes is randomly selected from a subset of the training images across all the classes, along the lines of Vidal-Naquet and Ullman (2003). A subset of these is then learned, including spatial-layout information, for a small set of image locations.

The model for learning the optimal features for sharing among which classes is based on boosted decision stumps, so named because they can be thought of as decision trees with just a single node, Torralba *et al.* (2007). Each weak learner selects a feature for a particular class, and then, at subsequent iterations, another class can only be added to the shared subset of classes for that feature if there is a suitable decrease in the classification error. Although the error for each shared class is higher than that for when a stump represents only one class, the total multi-class error is reduced because more classes have their error reduced by sharing.

As well as the overall type or types of features for the task, the appropriateness of individual features is a major consideration. One approach is to devise a small set of features or components which are deemed by the user to capture important aspects of the objects to be recognized, possibly based on theory of the type of constructs detected at different levels biological vision. For instance, Marr's 'generalized cylinders' allow the 2-dimensional, viewer-centred lower-level representations to be 'converted' to a 3-dimensional, object centred reference frame for recognizing deformable objects (Marr, 1982), while Biederman's alphabet of 3-dimensional 'geons' captures the 'non-accidental' properties of objects in images (Biederman, 1987). The problem with these structural shape primitives is the difficulty of reliably extracting them from images under different transformations (Ullman and Bart, 2004).

More recent approaches attempt to learn representative features directly from the training data. An initial set of features is extracted and refined through a process of feature selection.

3.4 Feature selection

Some systems generate a large number of features at the outset, many of which may be redundant or may be unreliable for the given task. For example, in medical diagnostic applications, only a relatively small subset of all the possible features might make good indicators for a particular disease or condition. Therefore it is vital that an optimal subset of features can be identified.

The main reasons for feature selection are:

1) To reduce the dimensionality of the data. A large number of features requires a lot of training examples, which may not be readily available. In very high dimensional spaces, for example, the feature space representing email messages in a 'spam'-detection problem (Janecek *et al.*, 2008), the data can be very sparsely distributed and if there are not enough training examples, it can be difficult for supervised classification systems, including neural networks, to converge well, especially since many of the features may be redundant or irrelevant. In unsupervised learning systems, a large number of dimensions can make the formation of distinct clusters difficult because the distances between data points tend to be more uniform than in a lower-dimensional space. This problem of high dimensional spaces is known as the 'curse of dimensionality' (Janecek *et al.*, 2008).

2) To allow better generalization in object classification and detection tasks. Having too many features can lead to overfitting of the data to the model, especially when there are relatively few training examples, for example, in a task of separating tissue samples from cancer patients from those of healthy subjects, the tissue biopsy data has thousands of variables but the sample only contains a hundred patients (Guyon and Elisseeff, 2003).

3) To speed up learning. Classifiers can be simpler with fewer, but more reliable features.

4) To facilitate better interpretation of system behaviour.

Feature selection techniques, in a supervised learning context, tend to either, weight or rank features in terms of their ability to predict a class, eliminating those that perform below a threshold, or, seek the optimal subset often from among many candidate subsets. Subsets can be evaluated by examining their ability to represent intra-class similarity and inter-class distinctiveness in the training data in schemes known as 'filter' methods. Another approach is to test the effectiveness of a candidate subset using a classifier in what are termed 'wrapper' methods. Yet other techniques 'embed' the feature selection process into the particular model, for example, decision trees and random forests (Guyon, 2003).

Because wrapper methods employ a particular classifier or model to test features, there is a danger that the model can become overfitted to the data, and in addition, assessing each feature subset on the basis of a classification attempt can be impractical if there is a large number of features, especially for online applications (Levi and Ullman, 2010). Filter methods are more efficient because of their direct application to the training data, but may result in less reliable representations of intra-class and inter-class characteristics. Two popular performance measures for filter approaches are mutual information and correlation.

Given that the initial feature set is often large, an exhaustive search for all potentially useful feature subsets is generally impractical, therefore more efficient search methods are generally employed. Common approaches to search are 'best first', 'greedy forward selection' and 'greedy backward selection (Guyon and Elisseeff, 2003), and also random search methods such as 'simulated annealing' and 'genetic algorithms' (Siedlecki and Sklansky, 1993). For wrapper approaches, these methods also reduce the problem of overfitting since fewer validation examples are needed than with exhaustive searches (Guyon, 2008). This is of particular benefit in applications, such as the medical example mentioned above, in which the number of potential features greatly exceeds the number of training examples. The use of random search in wrapper methods reduces the risk of getting stuck in local maxima.

3.4.1 Sampling

If the training set is large, it may not be practical to make use of the whole set to test out different subsets of features and sampling is often used to make the problem more manageable. Random sampling is quick but there is no guarantee that the examples selected will be optimal for learning good feature subsets. Another approach, known as 'active sampling' aims to choose instances that are informative for determining the relevance of features. Active sampling operates in two stages. First, the data is divided up according to some homogeneity requirement and then examples are randomly selected from the resulting partitions, as part of the process of 'active feature selection' (Liu *et al.*, 2003).

A convenient way to partition the data, if class labels are available, is according to the classes, or other homogeneity criteria can be applied to provide a finer partition based on closer similarities. Thus a dataset comprised of N examples distributed among C classes would be divided into non-overlapping subsets of $N_1, N_2, \ldots, N_C$ members. Random sampling can then be applied within each subset or '*stratum*' in a process known as '*stratified random sampling*'. The idea behind stratification is that the statistics of the data in a given stratum can be estimated by taking a small sample from it and then all the estimates can be combined to provide information about the whole dataset (Liu *et al.*, 2003).

3.4.2 Filter methods

Features may be selected on the basis of their relevance through a process of ranking, which involves assigning a score to a feature based on a function that represents some statistical measure of a feature's ability to predict the class.

Correlation criteria, such as Pearson's correlation coefficient (Guyon and Elisseeff, 2003), or normalized cross correlation, can provide a linear estimate of dependence between a feature and the target class. A high correlation score means the feature is relevant for the class. Weber *et al.* (2000) employ normalized cross correlation to detect and evaluate potential object parts in an unsupervised approach to learning constellation models for face and car recognition tasks.

Information theoretic approaches often employ an estimate of the mutual information between each individual feature and the class it is required to represent (Guyon and Elisseeff, 2003). Ullman and Bart (2004) define the usefulness of an image 'fragment' F to represent a class C as:

$$I(C, F) = H(C) - H(C|F) \tag{3.1}$$

where I is the mutual information that conveys how informative an image fragment, F, is about the class C, and H is the entropy, which indicates the degree of uncertainty. Thus the mutual information represents the decrease in uncertainty about whether an image belongs to the class C, given the occurrence of fragment F in the image. The expression for the mutual information can be written as:

$$I(C;F) = \sum_{c,f} p(C = c, F = f)\log\left(p(C = c, F = f)/\mathrm{p}(C = c)\mathrm{p}(F = f)\right) \tag{3.2}$$

The idea is to select the fragments with high mutual information for the representation. Mutual information is found to be optimal for features of intermediate complexity in terms of size or resolution (Ullman and Sali, 2000), and the aim of the feature selection approach is to produce a redundant set of overlapping features after dense feature extraction.

However, there is some disagreement as to whether redundancy in the representation is desirable and one of the main aims of feature selection is to eliminate redundant features (Janecek *et al.*, 2008). The problem with feature selection methods that rank features individually is that, although they select relevant features, they do not address the issue of redundancy (Guyon and Elisseeff, 2003).

In Ullman and Bart (2004), the aim is to maximize the amount of class information provided by a limited size of fragment set in which the members are highly independent. After each fragment has been assigned a mutual information score using the above equation, the system seeks to maximize the amount of class information provided by a limited size of fragment set in which the members are highly independent. This is achieved by an approximation to measuring the mutual information in terms of the joint distribution of the features in the selected subset. A

greedy iterative search approach is used. First, the fragment with the highest mutual information is added to the set and then each new candidate fragment, F, is assessed according to whether the information it provides about the class is contained in any of the fragments already selected. For each new candidate fragment, F, the most similar fragment to it, from those already selected, is found, and then the new fragment that provides the most additional information is added to the set. This process can be written as:

$$F_1 = \arg\ \max_F I(C;F);$$

$$F_{k+1} = \arg\ \max_F\ \min_i I(C;F|F_i) \qquad (3.3)$$

Estimating the input variable and class priors and joint probabilities using frequency counts, as above, works for discrete or nominal variables, but when the variables representing features and targets are continuous, a non-parametric method, for example Parzen windows, can be used to approximate the densities (Guyon and Elisseeff, 2003).

Another approach that evaluates features in the context of others is the *Relief Algorithm* (Kira and Rendall, 1992). This is based on the nearest-neighbour algorithm (Guyon and Elisseeff, 2003). The idea of Relief is to rank features according to how well their various values can separate similar examples of different classes. Figure 3.5 shows the original Relief algorithm for a two class problem.

Given m – number of sampled instances and k – number of features,

1. Set all weights $W[A_i] = 0.0$;
2. **for** $j = 1$ to m **do begin**
3. randomly select an instance X;
4. find nearest hit H and nearest miss M;
5. **for** $i = 1$ to k **do begin**
6. $W[A_i] = W[A_i] - \text{diff}(A_i, X, H)/m + \text{diff}(A_i, X, M)/m$;
7. **end**;
8. **end**;

Figure 3.5: The original Relief algorithm from Liu *et al.*, 2003, Figure 1.

For each randomly selected example, *X*, the basic Relief algorithm searches the training set for its two nearest neighbours, one from the same class, the nearest hit, *H*, and the other from the opposite class, the nearest miss, *M*. The weight for each feature is then updated with the distances between the example and each of its two chosen neighbours for that feature alone. The features with weights above a threshold value are then selected as relevant. The threshold can be determined statistically, so that the probability of rejecting a relevant feature as being irrelevant is below a required value decided by the user (Kira and Rendall, 1992). Alternatively, Kira and Rendall (1992) have found, by experiment, that there is a clear distinction between relevant and irrelevant features, with relevant features having positive weighting and irrelevant features having weights that are close to zero or are negative, so a suitable threshold can often be selected 'by eye'. Relief can also be applied in the context of multiple class problems (Robnik-Sikonja and Kononenko, 2003).

Kira and Rendall (1992) point out that one of the limitations of Relief is that it does not deal with redundant features. Guyon (2008) raises the issue that features that are irrelevant individually may have relevance conditional upon the presence of other features. This suggests that in ranking methods such as those that employ Euclidean distance, cross-correlation and mutual information, that only consider features individually, conditionally relevant features will be rejected. The 'conditional mutual information' approach of Ullman and Bart (2004) fails to tackle this problem, since it is not likely to select individual features with a low mutual information score as candidates for inclusion in the final fragment set. However, Guyon (2008) suggests that Relief does take conditional relevance into account in the fact that it uses all the features to compute the nearest neighbours, before evaluating the features individually.

3.4.3 Wrapper and embedded methods for feature subset selection

Both wrapper and embedded methods tackle the issue of feature dependency in that they select and evaluate features together in subsets. Wrapper methods use the classifier as a 'black box' that is retrained on every new feature subset (Guyon and Elisseeff, 2003). So they are very

simple method of feature selection and the use of greedy search methods such as forward selection or backwards elimination reduces computational complexity and overfitting.

Forward selection and backwards elimination each produce nested subsets of features. Forward selection gradually adds new features one-by-one and evaluates each new subset created, whereas, in backwards elimination, the process begins by evaluating the full feature set and then re-evaluates as features are removed one-by-one. Forward selection is the more computationally efficient approach, but at the cost of possibly producing less successful subsets than through the backwards method. The argument is that forward selection cannot assess the importance of features in the context of others that have not yet been included, and so the opportunity to retain variables that work well together to separate classes is reduced (Guyon and Elisseeff, 2003).

Belongie *et al.* (2002), in a wrapper approach, cluster training data to extract representative object exemplars and use a greedy cluster-splitting strategy to search the space of possible prototype subsets in order to select the optimum number of object class representatives. The next cluster to be split is determined on the basis of the overall misclassification error in a nearest neighbour classifier. The process terminates once the overall error drops below a certain level.

Using a wrapper method in an unsupervised setting, Weber *et al.* (2000) iteratively test small subsets of object parts to learn a probabilistic constellation model of an object class. After applying an interest operator to detect line intersections and centres of circular regions and a clustering step to further reduce the number of potentially useful parts initially extracted from these regions of interest, a greedy search method is used to learn an optimal set of parts. A few parts are selected at random to start with, and then the rest of the model parameters are estimated from the training images, using expectation maximization. At each iteration, an object part is substituted with a randomly chosen one and the model is relearned. Classification performance is then tested on a validation set of examples. If performance improves, the new part is retained. The process stops when there can be no further improvement.

Mutch and Lowe (2006) employ a backwards elimination wrapper approach to feature selection in a biologically inspired multiclass object recognition model which is a modified version of the standard model of Serre *et al.* (2005). The classifiers are binary linear SVMs each learning a separating hyperplane between two object classes, the members of which are d-dimensional vectors, with each dimension representing a feature. The elements of the d-dimensional normal vector to the hyperplane can be considered to be feature weights – the higher the weight value, the better the corresponding feature is at separating the two classes.

In embedded feature selection methods the search is guided by the learning process. It starts out with the full set of features and ends with an optimal reduced set. Using forward selection with trees, just a single path through is computed, by choosing at each split node the feature that yields the maximum reduction in entropy (Guyon, 2008).

Backwards elimination, in the form of recursive feature elimination (RFE), starts with all the features. A classifier is trained on the current feature subset by minimizing a 'cost' or 'risk' function. Next, for each feature, X_i, the change in the cost function is estimated, without retraining the classifier, for when that feature is removed. Then the feature, the removal of which most improves the value of the cost function, is eliminated (Guyon, 2008). This three-step process is iterated until a stopping criterion, such as no further improvement possible on the cost function or else a suitable feature set size, is reached.

Viola and Jones (2001) apply AdaBoost (Freund and Schapire, 1999) to select a small subset of features from a very large initial set at the same time as training the classifier. The AdaBoost algorithm is given in Figure 3.6 below. At each round of boosting, a weak learner is trained for each individual feature. Each weak classifier learns the optimal threshold for separating the positive and negative training examples with its single feature. The classifier with the smallest error is selected, the system weights are updated and that classifier's feature is added to the subset. The final strong classifier is based on a weighted sum of the selected weak classifiers.

Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where
$y_i = 0, 1$ for negative and positive examples respectively.

Initialize weights $w1,i = 1/2m, 1/2l$ for $y_i = 0, 1$ respectively,
where m and l are the number of negatives and
positives respectively.

For $t = 1, \ldots, T$:

1. Normalize the weights,

   $w_{t,i} \leftarrow w_{t,i} / \Sigma^n_{j=1} w_{t,j}$

   so that w_t is a probability distribution

2. For each feature, j, train a classifier, h_j, which is is
   restricted to using a single feature. The error is
   evaluated with respect to w_t

   $e_j = \Sigma_i w_i | h_j(x_i) - y_i|$

3. Choose the classifier, h_t, with the lowest error, e_t

4. Update the weights

   $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$

   where $e_i = 0, 1$ for example x_i classified correctly
   or incorrectly respectively, and $\beta_t = e_t / 1 - e_t$

   The final strong classifier is:

$h(x) = 1$ if $\Sigma^T_{t=1} \alpha_t h_t(x) >= 1/2 \Sigma^T_{t=1} \alpha_t$
$h(x) = 0$ otherwise
where $\alpha_t = \log 1/\beta_t$

Figure 3.6: The AdaBoost algorithm
from Viola and Jones, 2001, Table 1. Each round of boosting selects one feature from the 180,000 features initially extracted.

Another way to embed feature selection in the machine learning process is through the use of sets of decision trees. Ensembles of trees tend to more reliable than individual trees in object detection or classification because the decision made by the final classifier is based on the combined predictions of all the trees. The system of Maree *et al.* (2005) builds a large number of 'extremely randomized trees' or 'extra-trees' in which the thresholds for splitting at the internal nodes are chosen completely at random, rather than on the basis of a scoring mechanism. Each tree is then grown until it can correctly classify a labelled, randomly selected sample of image sub-windows, each represented by a vector of HSV colour space values for each pixel. After training, the sub-windows are discarded and for a test image, each of its sub-

windows is put through each tree and every tree outputs a prediction and the image is assigned to the class with the largest number of votes.

3.4.4 Feature construction

Some systems employ a clustering algorithm to 'construct' features (Guyon and Elisseeff, 2003). In this approach, sets of features that are similar in some way, for example, sharing the same shape or colour characteristics, are grouped together and represented by a single feature which is the cluster 'centroid'. The method is often unsupervised, but class information can be incorporated. The *K*-means and hierarchical clustering algorithms are often used. The aim of the *K*-means algorithm is to minimize within-cluster scatter and maximize between-cluster spread. The *K*-means algorithm iterates a two step procedure for minimizing a sum-of-squares clustering function:

$$J = \Sigma^{K}_{j=1} \Sigma_{n \in Sj} \ \|x^n - \mu_j\|^2 \qquad (3.4)$$

where K is the number of cluster centres that has to be decided out the outset, $\mu_j, j = 1,\ldots, K$, are the K vectors representing the cluster centres and are each the mean of the data points in one of the S_j disjoint subsets of N_j data points.

The data points are initially assigned randomly to K sets and then the mean of the points in each set is calculated. The next step is to reassign each point to the set which has the closest mean. The means are then recalculated. These two steps are repeated until either a set number of iterations is reached or there is no further reassignment of points (Bishop, 2002, p188).

Hierarchical clustering can be performed top-down – *divisive clustering* – in which all the features or data points are regarded as a single cluster and then clusters are split recursively on the basis of maximizing inter-cluster distance. Alternatively, in *agglomerative clustering*, each point is initially considered to be a cluster and then clusters are recursively merged on the basis the minimum inter-cluster distance (Forsyth and Ponce, 2003, p313).

Inter-cluster distance can be taken as the distance between the closest elements. This is single-link clustering. Another possibility is to take the inter-cluster distance to be the maximum distance between a member of one cluster and one from the second cluster in complete-link clustering. A dendrogram gives a visual representation of the cluster hierarchy and inter-cluster relationships, and enables the user to estimate the number of clusters (Forsyth and Ponce, 2003, p314).

Csurka *et al.* (2004) point out two important difficulties with *K*-means clustering – the algorithm tends to converge to local optima of the sum-of-squares objective function and the value of K must be determined by the user. The authors apply *K*-means to cluster SIFT patch descriptors to form a 'vocabulary' from which a 'bag-of-keypoints' representation is formed, to be used as a feature vector for image categorization. Despite awareness of methods that have been developed for automatically estimating the number of clusters, the authors choose an empirical approach, running the algorithm a number of times with different initial cluster centres and values of *K*, and selecting the clustering most likely to give the best image categorization results.

Jurie and Triggs (2005) produce a dense set of greyscale patch features extracted at multiple scales and then apply clustering to select a suitable subset, based on the idea that features that occur moderately frequently in images tend to be the most useful discriminators. Densely sampled patches in 'natural' images tend to be distributed in a non-uniform way as some tend to occur more frequently than others and the quantity of any particular texture in images is very variable. The authors explain that *K*-means clustering tends to form lots of clusters around the densest regions, with only sparse coverage elsewhere, due to the fact that the means tend to 'drift' towards the modes of the distribution as the cluster centre locations are iteratively updated. This problem is addressed through an 'online' clustering strategy in which the number of clusters does not have to be decided in advance. The algorithm employs 'mean shift', to position a new centre at the point of maximal density of a set of *N* uniformly and randomly selected unlabelled patches. The patches within a given radius of the centre are labelled and

eliminated. The process iterates until either the new clusters are no longer informative enough, or a sufficient number of clusters has been found.

Agglomerative clustering is used by Agarwal *et al.* (2004) as part of a class-specific vocabulary construction process. Each image patch is initially assigned to its own cluster and then pairs of similar clusters are merged if the average similarity between their respective patches is above a certain threshold. The similarity between patches is estimated using normalized correlation. This clustering sparsifies the representation, so that the number of parts selected is relatively small. However, when a cluster is labelled as a single 'feature', its constituent patches are retained and used as a redundant representation of a single 'conceptual' part with the aim of improving invariance to small changes in object part appearance. Part detection is on the basis of the maximum similarity of a part to a patch in the image. The similarity measure for a 'conceptual' part is derived from the average response, to the image patch, of a proportion of the most similar patches comprising that part, normalized correlation being used to compare patches.

3.4.5 Dimensionality reduction

While the main aim of feature selection is to eliminate irrelevant and redundant features, the technique of 'dimensionality reduction' is used to reduce the size of the feature space to manageable proportions without losing much of the information it originally contained. No information about individual features is lost during feature selection, but it may sometimes be necessary to discard potentially useful features if the application can only make use of a small set of features. On the other hand, with dimensionality reduction, information on the contribution of individual features is usually lost, as the new components are created from linear combinations of the original features (Janecek *et al.*, 2008).

Two frequently used techniques for dimensionality reduction are Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA). PCA finds the dimensions that are associated with the largest variance in the data distribution, without taking class labels into

consideration, whereas LDA determines the decision boundaries that best discriminate among the classes (Martinez and Kak, 2001). PCA uses a linear transformation that maps the original feature space, usually, onto a lower dimensional one. The resulting feature vectors are in the form:

$$y_i = W^T x_i \qquad i = 1,\ldots, N \tag{3.5}$$

where the columns of the matrix W are the eigenvectors, $\mathbf{e}_i$, obtained by solving

$$\lambda_i \mathbf{e}_i = \mathbf{Q}\mathbf{e}_i \tag{3.6}$$

where $\mathbf{Q} = \mathbf{X}\mathbf{X}^T$ is the covariance matrix and λ_i is the eigenvalue associated with eigenvector $\mathbf{e}_i$.

The aim of LDA is to maximize between-class distance and minimize the within-class variance. In Fisher's linear discriminant this is achieved by maximizing the ratio of the two variances, which is known as the Fisher criterion (Bishop, 2006, p188).

Martinez and Kak (2001) compare PCA and LDA on a face discrimination task and find that, overall, that while LDA can out-perform PCA on larger datasets, PCA can be more reliable than LDA when the dataset is small and that PCA performance is more consistent across different datasets, Abstract. As an example of the problem of LDA with small datasets, Martinez and Kak (2001) draw attention to some of the results of a competition on the FERET face recognition database. It was found that the results of experiments by Turk and Pentland, (Turk and Pentland, 1991), using PCA, were better than those obtained in an experiment conducted at the University of Maryland, using LDA, where in both cases only a small number of examples per class were available.

PCA is often applied to reduce the number of dimensions to be considered when building probabilistic representations. For example, in the work of Fergus *et al.* (2003), several object classes are each represented with a generative, probabilistic constellation model. The features that describe the object parts are found using a saliency detector over location and scale and a subset of the highest-scoring regions is selected. PCA is applied to reduce the dimensionality of

the features in order to make the parameters of the Gaussian appearance densities easier to manage.

3.4.6 Feature selection in 'scene-to-sound' mapping

Another area of research in which large numbers of scene features can create considerable problems is that of converting visual scene information into a sound signal to enable blind users to interpret their surroundings. Standard optophonic mappings transform points that are higher up in a scene to higher-pitched sounds, and for brighter regions the sounds are louder and horizontal position is linked to time, as images are scanned left-to-right. Capp and Picton (2000) introduce depth information into the model and use loudness to represent proximity to the user, which means that brightness and colour information about the image is lost. A depth map is generated from the images obtained from two cameras. The system makes use of 'top-down' information about visual perception of objects in depth, in that objects that are nearer to the user are displayed as brighter than those that are further away, with louder sounds being associated with brighter regions in the optophonic transformation. This greatly reduces the amount of information the user must interpret and provides the same sort of information as an ultrasonic system, but the processing is slower. Another major problem with this approach is that a page of text cannot be displayed, because the surface with the text on it is flat and so the characters cannot be detected as being at a different distance from the user than the background. This issue is addressed through the generation of 'edge depth maps' in which edge detection information is incorporated in order to show the outline of objects including characters in the image.

Edge depth maps can be difficult to interpret even for simple scenes especially if edge information is missing. A further improvement to this type of representation is made in Picton and Capp (2008). An existing 'cartoon' image technique is applied to the edge depth map to fill in shading in the regions between the edges. This approach further reduces surplus information and is intended for use with any existing stereo scene mapping system.

3.4.7 Mapping to higher dimensions

Although many machine vision systems seek to make the representation of the image data more manageable by reducing the number of dimensions, some are employing a more biologically-based approach of a sparse, overcomplete representation, in which the feature vector has more dimensions than the input, but only a small number of elements are 'active' for any particular image (Ranzato *et al.*, 2006). The system designed by Ranzato *et al.* is based on minimizing energy based on squared distance, in an encoder-decoder model in the process of learning an optimal code vector for a given input image patch. Learning is achieved by first finding the optimal set of codes for a given set of filters in the encoder and the decoder and then fixing these codes while updating the system weights in an iterative process, that minimizes the encoding prediction energy of the encoding stage together with the reconstruction energy in the reproduction of the input patch in the decoding stage. The reconstruction is attempted using the sparse code derived from the sparsifying logistic module which sparsifies the code vector by applying a type of weighted softmax function to limit the frequency and duration of the activity of each individual processing unit, over the training samples.

Evaluating the similarity between patterns can be difficult in high-dimensional spaces, for example in computing the dot-product for SVMs.

The 'kernel trick' for distances involves using a suitable kernel for determining in a non-linear way the similarity between pairs of training patterns in the lower-dimensional space, which is the equivalent of evaluating a dot-product in a higher-dimensional space (Scholkopf, 2001).

The disagreement over the benefits of sparse versus dense representation is discussed in Section 3.4.8 of the thesis.

3.4.8 Dense versus sparse representations

As in biological vision research there is some debate as to whether the primate visual system makes use of a sparse, compact or dense representation at various levels of the visual hierarchy,

in machine vision research there is some disagreement as to whether a dense or a sparse representation is more effective.

Jurie and Triggs (2005) advocate a dense representation, having found that large codebooks of features generally performed better than reduced sets, as more discriminative information is retained in dense codebooks. On the other hand, in Mutch and Lowe (2006), the principle of optimum sparseness is followed with a view to enhancing the generalization ability of the system. Which system operates better may depend on the number of classes to be discriminated – fewer classes requiring less sparseness. Jurie and Triggs classify fewer classes than does Mutch and Lowe. It may also depend on whether the representation is shallow or multilevel. Jurie and Triggs use a single-layer, 'bag-of-features' approach, whereas Mutch and Lowe's is a multilevel system. Another important factor is how much visual detail is required for the task. If the task is to recognize a face regardless of facial expression, then selecting just the lower-frequency components in the image intensity signal is likely to be the desired approach (Lai *et al.*, 2001), whereas a fuller representation that also includes the higher-frequencies captures the more subtle variations required for discrimination of different facial expressions.

Agarwal and Triggs (2006) explore the concept of a trade-off between base-level codebook size, that is, the number of low-level features, and the number of levels of processing required. Although the finding is that having a larger codebook at any level tends to improve performance, there is some indication that having more levels is more effective than increasing the number of features at the lowest level.

Sparse representations reduce the capacity of the classifier as fewer patterns can be stored. The ultimate sparse system would only activate one output per class, as in the 'grandmother' neuron analogy (Rolls and Deco, 2002, p11).

3.5 Generative versus discriminative systems

Tu (2007) explains that generative models attempt to learn the underlying distribution of the data, while discriminative models try to define the boundaries between different classes for classification purposes.

3.5.1 Discriminative models

In a *discriminative* model, a classification decision, that minimizes the probability of misclassification, can be made by assigning a pattern to class C_k if, given the input feature vector, x, the probability that the correct class is C_k is greater than the probability that it is C_j, that is:

$$P(C_k|x) > P(C_j|x) \text{ for all } j \mathrel{!}= k, \tag{3.7}$$

where $P(C_k|\mathbf{x})$, $(1 <= k < = N)$ are the posterior probabilities of the distribution of the d-dimensional vectors $\mathbf{x}$ over the pattern space and N is the number of classes. Using Bayes' theorem the posterior probabilities can be derived by combining the prior probabilities $P(C_k)$ and the unconditional density $p(\mathbf{x})$ with the class-conditional densities, $p(\mathbf{x}|C_k)$ to give:

$$P(C_k|\mathbf{x}) = p(\mathbf{x}|C_k)\, P(C_k)/p(\mathbf{x}) \tag{3.8}$$

where $p(\mathbf{x}|C_k)$ represents the probability that, given class C_k, the vector x will occur, and $p(\mathbf{x})$ is acting as a normalization factor. The priors and the class-conditional densities are easier to estimate than the posteriors and so the decision rule in equation (3.7) above can be expressed as

$$p(\mathbf{x}|C_k)\, P(C_k) > p(\mathbf{x}|C_j)\, P(C_j) \text{ for all } j \mathrel{!}= k \tag{3.9}$$

It is often pictured that the feature space or pattern space is divided into decision regions, with the divisions between these regions referred to as decision boundaries or decision surfaces. A classifier attempts to minimize error by setting the decision boundaries so that a data point, x, falls in the region associated with its true class, which is the equivalent to assigning x to the class with the largest probability of being correct as in (3.9).

This gives rise to the notion of a discriminant function $y_k(\mathbf{x})$ which can be determined by the training data directly, without the necessity for probability density estimation.

The classification approach just described does not allow the option to reject an example if the decision output is not strong enough. For example, if a quality control system is bound to classify items as either satisfactory or not, then if the larger output from the discriminant function is not very high, the system has to classify a borderline quality item as satisfactory or not, when it would be more appropriate for a human operator to intervene. If a suitable threshold is set, the decision becomes:

If $\max(y_k(\mathbf{x})) >$ threshold, assign sample to class C_k, else reject (Bishop,2002, p28).

A rejection threshold can be set with the aim of minimizing the expected 'loss' or 'cost' of a misclassification, while also taking into account the cost of a rejection requiring to be checked by a human operator. Using expert knowledge of the task in hand, a rejection loss value, L_r, can be set, and a KxK loss matrix can be defined, the elements, L_{kj}, of which represent the loss associated with assigning sample $\mathbf{x}$ to class C_j when the true class is C_k, whether or not $j = k$. For instance, in the above example, it might be decided that there should be a greater loss associated with classifying a faulty item as satisfactory, than with misclassification of a satisfactory item, while correct classification would be expected to incur no loss.

Then, to minimize the expected loss, a new sample, $\mathbf{x}$, is assigned to the class, j, for which the value:

$$\Sigma_k L_{kj} p(C_k|\mathbf{x}) \tag{3.10}$$

is at a minimum, provided that the minimum value is less than L_r, else the sample is rejected (Bishop, 2006).

Examples of discriminative models are: support vector machines (SVMs), Boosting and Neural Networks.

3.5.2 Generative models

In a generative model, the probability density function of a finite set of data-points drawn from that function is estimated. There are two main approaches, a parametric method, in which a particular class of function is selected to model the data – often a Gaussian distribution is used, as this gives a good representation of data that tend to cluster around a mean value.
Other examples of generative models are Gaussian mixture models, multinomial distribution, hidden Markov models and Naïve Bayes, Latent Dirichlet Allocation and probabilistic Latent Semantic Analysis.

Once the form of the density function has been decided, the parameters for fitting the data to the chosen model are learned. There are two approaches to this – Maximum Likelihood and Bayesian Inference (Bishop, 2002, p39).

Maximum likelihood estimates the values of the mean and covariances of the data among other parameters (Bishop, 2002, p41), whereas the Bayesian approach does not set the values of the parameters, but instead, models the uncertainty in the parameter values by a probability density function. The parameters are broadly represented by a prior probability density, then once the data have been observed, Bayes' theorem is applied to determine the posterior probability density. The posterior probability narrows down the uncertainty in the mean of the data, (Bishop, 2002, p42).

Fei-Fei *et al.* (2007) employ Bayesian inference to learn a generative model of appearance and spatial layout of objects using only a few training examples. Learning is done using the Variational Bayes method to approximate the posterior distribution of the model parameters. The algorithm iteratively updates the hyper-parameters and hidden variables to minimize the difference between the actual posterior distribution and its approximation. Since prior probabilities are incorporated in the model, it means that information about the statistics of previously learned classes can be used in learning the new classes, which cannot be done with a maximum likelihood approach. In comparison with the maximum likelihood-based model of Fergus *et al.* (2003), the Bayesian approach is found to require fewer training examples and it

outperforms the maximum likelihood model on small datasets. The work of Fei-Fei *et al.* (2007) illustrates the point that the choice of prior can affect performance. The authors point out that the simple prior used in the model does not capture enough information about the variability in the training data. Section 3.6 of the thesis discusses this further.
Maximum likelihood does not rely on prior knowledge which can be an advantage when such information is not readily available, however, an example given in Bishop (2006, p23) illustrates the value of including prior information in the model. A fair coin is tossed three times, and each time comes up heads. The maximum likelihood estimate the probability of coming up heads as '1' for all future throws, whereas the incorporation of the prior probability of 0.5 for heads, for example, into a Bayesian model would give a more reasonable estimate.

Both the Bayesian model of Fei-Fei *et al.* (2007) and the maximum likelihood model of Fergus *et al.* (2003) provide rich, complex representations of objects, through fully-connected constellations of object parts, the model of Fergus *et al.* being the more complex as it takes occlusion of parts into account. While this complexity provides a lot of information about the appearance and spatial layout of the objects being represented, the number of parts that models can handle is relatively small – between 3 and 7 (Fei-Fei *et al.*, 2007).

Csurka *et al.* (2004), on the other hand, offer a rather simpler model for object categorization, based on a bag-of-words representation. Bag-of-words models originate in the field of document categorization. The 'words' are referred to in this paper as keypoints. The model derives keypoints from the centres of clusters of feature vectors representing image patches. This allows a larger number of parts or patches to be considered. An image is represented by a histogram of the occurrences of each of its constituent keypoints. This type of 'shallow' representation is discussed further in Section 3.8 of the thesis. A generative and a discriminative approach to categorization are compared.

The generative approach employs a Naïve Bayes classifier for which the assumption is that the conditional distributions for the keypoints are independent, so that Bayes' rule can be applied in the form:

$$P(C_j|I_i) \propto P(C_j)P(I_i|C_j) = P(C_j) \prod_{t=1}^{|V|} P(v_t|C_j)^{N(t,i)} \quad (3.11)$$

where I_i is the image in question, v_t is the current keypoint and N(t, i) is the number of occurrences of keypoint v_t in image I_i, and the largest a posteriori output is taken to indicate the classification prediction.

In the discriminative method, a set of m linear SVM classifiers is learned, each determining a hyperplane that separates a particular object class from the other $m - 1$ classes with a maximal margin, the margin being the distance of the nearest training data point from the hyperplane. A test image is assigned to the class with the largest classifier output. The corresponding classification function is given by:

$$f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b}) \quad (3.12)$$

where $\boldsymbol{w}$ and $\boldsymbol{b}$ are the hyperplane parameters.

The results show that the SVM performs better than the Naïve Bayes on seven object classes. This could be because the bag-of-words model used here is too simple to represent the variability in the image data effectively, or that the SVM system is able to 'ignore' that variability and concentrate on finding reliable decision boundaries. Ulusoy and Bishop (2005) make the point that the details of the data distribution that are modelled by a generative approach may not be relevant for determining the a posteriori probabilities.

The more complex model of Sivic *et al.* (2005) uses probabilistic Latent Semantic Analysis, (pLSA) to discover topics or object categories in unlabelled data and to classify and detect and loosely segment out objects in images. As with Csurka (2004) the underlying approach is a bag-of-words model that ignores spatial information. A vocabulary of 'visual words' or object parts is learned by k-means clustering of the SIFT descriptors of image patches and using the cluster centres, pLSA is employed in learning to represent each image as a mixture of four topics. Each topic is represented by a histogram of the occurrences of each word in the vocabulary and each document or image is modelled by a histogram made up of the histograms

for each topic. Maximum likelihood is used to estimate the model parameters through maximizing an objective function.

Tu *et al.* (2007) employ discriminative approaches to learn generative models for a variety of visual tasks including texture classification and face modelling, Section 4. The goal is to end up with a set of pseudo-negative examples from a reference distribution which are indistinguishable from the positive training set. The process is iterative, with a new reference distribution of pseudo-negatives being generated recursively by bootstrapping or sampling at each round, and these samples being used along with the positive training set to train a classifier through boosting. The training error increases on each round, as the pseudo-negative distribution becomes increasingly similar to the positive training set. The process stops when the training error exceeds a threshold, indicating that the two distributions are practically indistinguishable.

Thus the discriminative aspect helps with learning the generative model of the positive training data and the generative aspect provides the negative training examples generally required by the discriminative approach, increasing its modelling ability and improving the decision boundaries.

3.5.3 Non-parametric methods

Non-parametric approaches do not attempt to model the distribution of the data as a whole, but instead to use local estimates. Examples of this approach are histograms, *K*-nearest neighbours and kernel-based techniques.

Histograms have the advantage that the data can be discarded once the histogram has been made, unlike *K*-nearest neighbours and kernel-based methods for which the training data is needed for estimating the density of new data instances. Selecting an optimal number of bins can be a problem with histograms, since the number of bins determines the smoothness of the resulting density estimate. This problem can be helped, for example, by applying clustering to the task of determining the ranges of the bins. Another difficulty is that the number of bins grows exponentially with the dimensionality of the feature space (Bishop, 2002, p51).

With kernel-based density estimation, the kernel width is fixed and the density is estimated from the proportion of data points falling within that fixed region. Both histogram and kernel-based methods have the problem of discontinuity at the edges of the bins or windows which are not representative of the true underlying distribution of the data (Bishop, 2006, p121). Using a smooth kernel function, such as a Gaussian, reduces the effect of the discontinuities associated with the 'hypercube-based' kernel, and gives a smoother density estimate (Bishop 2006, p123). Again, the kernel width is critical for how smooth the resulting density estimate is. Too wide a kernel will tend to over-smooth the estimate, losing important information, while a narrow kernel can result in a noisy model (Bishop, 2006, p124).

Instead of fixing the volume of the hypercube and counting the number of data points that fall within it, the number of data points can be fixed and the volume of the sphere centred on a particular point allowed to grow until it contains the required number of points. This is the approach of *K*-nearest neighbours. In this approach it is the number of data points that controls the amount of smoothing (Bishop, 2006, p125). A *K*-nearest neighbour classifier – assigns a test instance to the class with the largest number of data points among the set of *K*-nearest neighbour training examples.

For large datasets, the search for nearest neighbour among all the training examples requires costly computation, but this can be reduced if efficient tree-based search methods are employed (Bishop, 2006, p126). Another way of reducing the search is to 'edit' the training data, as discussed in Section Feature Selection. Belongie *et al.* (2002), do this by clustering to find suitable prototypes. Provided the amount of data to be processed can be kept to a manageable quantity, the advantage of this type of classifier is that training only involves storing the training set. Another advantage is the fact that this kind of instance-based learning approach is that the local approximation of the target function when classifying a new instance avoids the need for a complex representation to be learned for the whole distribution (Mitchell, 1997, p231).

Belongie *et al.* (2002), employ a weighted version of the distance measure in a *K*-nearest neighbour classifier for hand-written numeral classification, so that closer neighbours to the test data point influence the decision more strongly than more distant ones.

3.6 Learning from few examples

A problem with systems that are able to detect and classify a large number of different categories of object is that they generally need a considerable amount of training data for each class. In many types of visual task there is little data available, for example, for medical diagnostics representations are often high-dimensional but data is relatively scarce. If examples are in short supply, one solution is to generate more from the existing data by performing small image distortions or adding noise, or to learn within-class variations through a generative model. However, even if sufficient data can be made available, having to learn a representation from a large quantity of examples is very inefficient and is not feasible for online applications. Primates can learn new types of object or different instances of the same category apparently effortlessly, adding, for example, a new face to the repertoire with sometimes just a single viewing (Rolls and Deco, 2002, p120). Hence techniques that might enable machine vision systems to learn from just a few examples are being extensively researched.

Torralba *et al.* (2007) advocate training detectors on multiple objects, with shared features, rather than on individual classes as a way of reducing the amount of training data required. It is shown that the performance of single-class classifiers tends to be adversely affected when only a small number of training examples is available. However, this approach does not accommodate the introduction of a new class, since the optimal way of sharing the features is learned across all the classes at the outset.

A technique referred to as "cross-generalization" devised by Bart and Ullman, is an approach to learning new classes from a single example by using features associated with already-learned classes to select features for the new class (Bart and Ullman, 2005). The features used are class-specific image fragments. The idea is to select features in the new class example that most closely match informative features from familiar classes that are similar to the new class. No

negative training examples are required in learning the novel class, which mimics primate learning of new objects.

Fei-Fei *et al.* (2007), in a generative probabilistic model, incorporate prior information from three unrelated object categories in the Cal-tech 101 database, faces, spotted cats and aeroplanes into a Bayesian incremental algorithm for learning new classes with just a small number of training examples. The authors acknowledge that the limited number of object categories employed to derive the prior makes its contribution to modelling the new classes relatively weak. Bart and Ullman (2005) emphasize the importance of the similarity to the novel class of familiar classes used in learning a suitable representation of the new class.

Opelt and Pinz (2006) present a model similar to that of Torralba *et al.* (2007) in the joint learning of classes and sharing of features, except that the system builds a common "alphabet" of boundary fragments derived from a set of object categories that can be used to represent all subsequent categories. Boundary fragments can be shared in three different ways. If a new boundary fragment is sufficiently similar to a stored alphabet example from a different object class, the stored entry's list of classes it represents is updated. Fragments can be tested on all the object-category validation sets with suitably close matches indicating that they are representative of the corresponding classes. The third sharing circumstance is when a fragment is a close match for a particular class on the validation set, but its relationship to the object centroid is different.

In addition, weak detectors are shared among strong detectors in an incremental Joint-AdaBoost learning scheme. When the system is presented with a new class, the existing weak detectors can be tried and if any are suitable, they can be reused in forming a new strong detector for the category, thus reducing the number of new weak detectors to be learned.

The approach of Thrun (1996) is again based on the idea of the ability of humans to draw on past experience when learning a new task in a framework of "lifelong learning". An example given by Thrun is that of learning to recognize a new person with the help of invariant features such as eye-shape as opposed to factors like facial expression. The research demonstrates that

previously learned examples, referred to as 'support sets' can be used in the process of learning an intermediate representation, which can then be employed by a subsequent classifier to improve generalization ability when only a few training examples are available, (see also Wolf *et al.*, 2006). The data in the support sets may be either labelled or unlabelled depending on the learning approach employed.

The technique of making knowledge from previously learned tasks available for helping to learn new ones is known as "transfer learning". Quattoni *et al.* (2008) use unlabelled data and labelled training sets from related problems to learn a sparse set of prototypes for training a classifier for a particular task, in this case, predicting whether an image belongs with a given news topic. A prototype representation of the unlabelled data is derived first and then a subset of the prototypes is selected with the help of the training data from the related problems, and finally a new representation is created using the kernel distances to these prototypes.

Levi and Ullman (2010) introduce an efficient method of updating the feature set representing a particular class of object in an adaptive online feature selection scheme. The problem addressed is that, given new instances of a class over time, the characteristics of the class can change. In order to accommodate the variation in a new example, the approach is often to add a new set of features to the class representation, but in a task such as hand-writing recognition, this could lead to an unacceptably large set of features. The approach here is to gradually adapt the feature set to cope with evolving variation while keeping the representation to a manageable size.

Initially a large set of features is extracted from the training set and then the online algorithm is gradually fed class exemplars and a small subset of features in an iterative process, during which a new feature is evaluated in relation to the current set of features in terms of the amount of class information it can provide, relative to the other features in the set, about a continuously updated fixed-size set of recent training examples. Informative features can then replace features that are no longer so relevant.

3.7 Image Segmentation

An important aim of image segmentation in machine vision applications is to separate out foreground regions or objects from the background. It has many applications, including medical imaging, to locate tumours, for example; interpreting satellite images, for instance to find forests or water; and object detection and recognition, for example in face recognition and fingerprint identification.

There is evidence from research, that, in human vision, object recognition may precede and facilitate image segmentation, rather than the other way round, (Wolfe, 1996). Traditional methods of image segmentation in computer vision tend to be based solely on bottom-up processing, which is more akin to the rapid parallel search mode of the visual attention mechanism. However, feed-forward image segmentation techniques do not necessarily achieve segmentation of whole objects from the background. The result of a segmentation algorithm is a set of segments that cover the whole image, or a set of contours if edge detection is employed and a model is then fitted (Forsyth and Ponce, 2003, p329), for example, through the application of a form of the Hough transform, to find lines, circles or more general shapes.

3.7.1 Clustering pixels to form image segments

Pixels can be grouped together according to certain characteristics or requirements, such as colour, intensity, texture, proximity. As in the concept of feature construction (introduced in Section 3.4.4 of the thesis), clusters can be formed using algorithms like *K*-means, or by means of hierarchical divisive or agglomerative approaches. A problem with divisive and agglomerative clustering is that an exhaustive segmentation is generally not practical due to the number of pixels in an image. Instead, the process has to be stopped using a threshold of some kind – in divisive clustering a limit might be set on the number of clusters while in agglomerative clustering, segmentation might stop once the inter-cluster distance becomes sufficiently small.

Another difficulty with the large number of pixels is in searching exhaustively for the best split or merge at each iteration. In divisive approaches, a histogram of pixel colour in a region is one possible way to indicate a reasonable split. In agglomerative methods, generally only clusters that share part of their boundaries are merged. Also, regions tend to merged on the basis of being sufficiently close, rather than necessarily being the closest.

K-means obviates the need for determining merging or splitting criteria, but there is the problem of having to decide on the value of *K* and in addition to this, because the algorithm does not make use of any information about pixel location or local texture, the resulting segments are often disjoint and scattered (Forsyth and Ponce, 2003, p315).

3.7.2 Histogram-based segmentation

Histogramming is often used in the process of image thresholding. The idea is that, when suitable thresholds can be found, an image can be segmented into objects and background. If an image histogram has several modes, each one can be considered to approximately correspond to a region in the image, with the valley between adjacent modes containing a potential threshold point. One way of estimating optimal threshold values is by means of hierarchical cluster analysis (Arifin and Asano, 2006). The approach in this work is to 'build' a histogram, starting with a bin or cluster assigned to each grey level in the image. Two adjacent clusters are then merged on the basis of the difference between their means and the variance of the new cluster resulting from the merge. Clusters are labelled and the highest greyscale value in each cluster is stored as a potential threshold. At each iteration of the merging algorithm, the clusters are relabelled and thresholds are reassigned since the number of clusters has decreased. The algorithm can terminate when the desired number of clusters is reached and the threshold estimates are the highest greyscale value of each remaining cluster. Thus it is a multilevel approach. Unless the aim is simply to binarize the image, the problem is, again, deciding the number of greyscale levels that will give a meaningful segmentation.

Most image thresholding techniques use direct information about individual pixels, such as their own greyscale or colour value, or its relationship to that of pixels in the immediate neighbourhood, to form clusters or histograms. Johnson and Simon (2001) take a different approach. In this work, fundamental primitive structures are defined in greyscale images. Such a structure is a set of pixels, $p_{x,y}$, $p_{x+1,y}$, $p_{x+2,y}$,....., where the greyscale value, $g_{x+i,y} >= g_{x+i+1,y}$, with the higher values being brighter, Figure 3.7 (adapted from Johnson and Simon, 2001, Figure 8).

Figure 3.7: A gradient run primitive
adapted from Johnson and Simon, 2001, Figure 8

When considering the 4-neighbours of a pixel, four types of these gradient runs of pixels can be found in digital images, left-to-right, right-to-left, top-to-bottom and bottom-to-top. Gradient polygons are formed from contiguous sets of gradient runs, Figure 3.8 (adapted from Johnson and Simon, 2001, Figure 9).

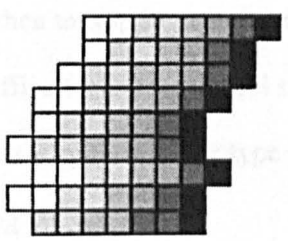


Figure 3.8: A gradient polygon
adapted from Johnson and Simon, 2001, Figure 9

Using the idea of gradient, a new type of histogram is formed. Associated with each gradient run is a pair of numbers – g_{light} and g_{dark}. These values can be thought of as defining a point in a 2-dimensional grid with one of the dimensions, L, representing g_{light} and the other, D, representing g_{dark}. At each location on the grid, the number of runs for which g_{light} = L and g_{dark} = D can be stored. These run-counts can then be plotted as a 3-dimensional histogram showing the distribution of the runs. The application in this work is the reading of hand-writing on bank cheques. Thresholding is applied to provide a segmentation of the whole image into an initial

set of regions, from which histograms representing more local and coherent subregions can be generated. Histograms can be computed for horizontal, vertical and diagonal runs, but only horizontal runs are used here. The peaks in the initial histogram, represent different types of region in the cheque. An initial segmentation threshold is chosen by separating off the peak representing the lightest pixels. Then to separate the writing out from the relatively dark filigree pattern in the top left of the cheque, the histogram of this region is then thresholded to separate out the darkest peaks which are associated with the hand-writing in this region, Figure 3.9 (from Johnson and Simon, 2001, Figure 19).

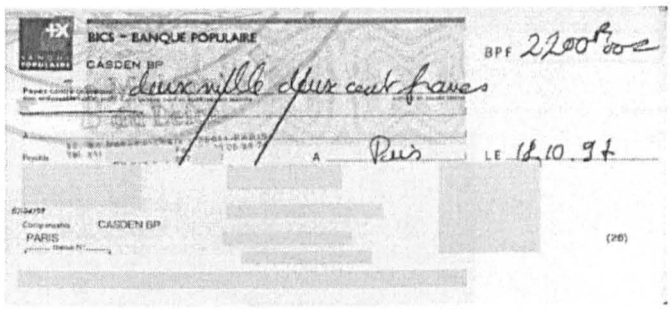


Figure 3.9: Cheque with hand-writing in a dark filigree pattern region from Johnson and Simon, 2001, Figure 19.

Again there is the difficulty of when to stop segmenting in general applications, since when regions become too small it is difficult to extract useful statistics from them. As the authors point out, this histogram approach is applied to one type of task, making it easier to employ domain knowledge in engineering the system.

3.7.3 Graph-theoretic segmentation

Another approach to segmentation is to represent an image by an undirected graph, where the nodes represent points in the feature space and an edge connects each pair of nodes. An edge is weighted according to the degree of similarity between the nodes it connects. The idea is then to, recursively, determine suitable points at which to 'cut' the graph, by removing edges, so that the resulting decomposition of the global graph into sub-graphs forms a good segmentation of the image. If a graph $G = (V, E)$ is partitioned into two disjoint sets of nodes, A and B, with $A \cup B = V$ and $A \cap B = \emptyset$, the dissimilarity between the two parts can be estimated by the sum of the weights on the edges that have been removed:

$$cut(A, B) = \Sigma_{u \in A, v \in B} w(u, v) \quad (3.13)$$

(Shi and Malik, 2000, equation 1)

This is known as the minimum cut criterion. The best split of the graph is achieved when this value is at a minimum. This process of minimizing the cut value to find the best split the current segments can be applied recursively until the required number of segments has been produced (Shi and Malik, 2000).

However, the authors point out that this method can have a tendency to partition off isolated nodes from the graph, Figure 3.10.

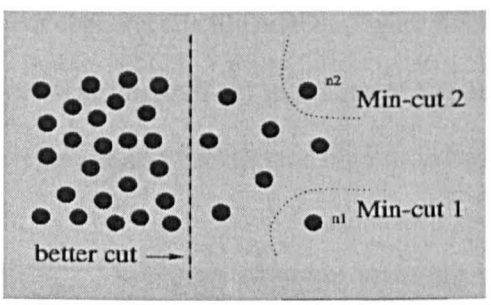


Figure 3.10: The minimum cut can give a bad partition
Shi and Malik, 2000, Figure 1.

Figure 3.10 illustrates how this can happen. Given that the weights on the edges are inversely proportional to the distances between nodes, the cut that segments off node n1 or n2 from the rest of the graph will have a value that is the sum of a lot of small weights rather than a sum of larger weights that would result from the more central split.

To tackle this problem, the authors introduce the *normalized cut* (*Ncut*):

$$Ncut(A, B) = cut(A, B)/assoc(A, V) + cut(A, B)/ assoc(B, V) \quad (3.14)$$

(Shi and Malik, 2000, equation 2)

where $assoc(A, V) = \Sigma_{u \in A, t \in V} w(u, t)$ is the total connection from the nodes in A to all the nodes in the graph and $assoc(B, V)$ is the equivalent for the set B.

This computes the cost of a particular cut as a fraction of the edge connections to all the graph nodes, which means that, for example, for segmenting off node $n1$ in Figure 3.10, the *cut* value is 100% of the sum of all the connections from that node, thus reducing the bias toward that partition of the graph. The optimal value of the normalized cut is determined by solving a generalized eigenvalue problem at each iteration of the graph-partitioning algorithm.

3.7.4 Top-down segmentation

The segmentation systems discussed so far have largely only made use of bottom-up processing. Borenstein and Ullman (2002) employ high-level, class-specific information to separate objects from their backgrounds. The difficulty with segmenting images without the benefit of prior knowledge of the object concerned is illustrated. Figure 3.8, (from Borenstein and Ullman, 2002, Figure 3) shows the normalized-cut technique of Shi and Malik can cause objects to be split into parts and foreground and background regions to be merged.

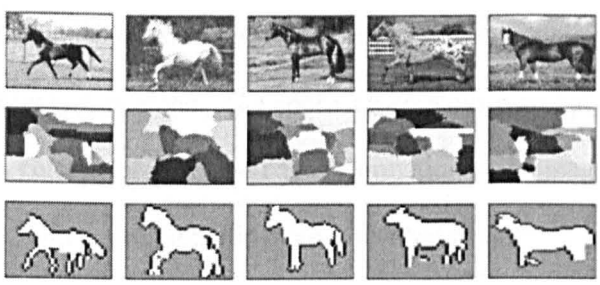


Figure 3.11: Segmentation by normalized cut compared with Borenstein and Ullman algorithm
Examples of horse images (top row) segmented into subregions by the normalized-cut algorithm (middle row) and segmented into figure-ground by the algorithm of Borenstein and Ullman. From Borenstein and Ullman, 2002, Figure 3.

Borenstein and Ullman's segmentation scheme is applied to a single class of object, namely side views of horses facing to the left, but with significant within-class variability. The class is represented by a pool of fragment primitives for the shape, each stored as a template, along with a figure-ground label and 'reliability' score. The algorithm searches for an initial set of fragments most likely to give an accurate partial cover of the object, based on the reliability score. The full cover is then completed by the addition of less reliable fragments. The process

of finding an optimal cover is iterative and attempts to maximize the match quality, consistency and reliability of constituent fragments.

Individual fragment matching is done using just the figure part of the figure-ground template, to reduce background noise effects, but with the addition of an edge detector to provide boundary information. The authors suggest that their class-based segmentation approach could benefit from being combined with traditional image-based segmentation, so that the image-based aspect could provide information on salient regions in which objects might be located.

3.7.5 Combining segmentation with recognition

Shotton *et al.* (2009) combine image segmentation and object recognition in the same process, Section 1.1. Sometimes object classes share similar parts, causing appearance-based object recognition and image segmentation to be unreliable. The example given here is windows in cars, planes and buildings. The model in this work incorporates appearance, spatial layout, and context information from the surrounding image, to reduce the ambiguity that can arise from the use of appearance information alone. The approach employs a probabilistic framework, or representing texture, colour, location and edge information in an image segmentation and classification model. Often, probabilistic models attempt to define a joint distribution over observation sequences X and class labels Y, in the form $p(X, Y)$, in which all possible combinations of observation sequences and labels must be represented. This is generally not practical unless the elements of each sequence are assumed to be independent of each other, which is generally not a realistic assumption. An alternative approach is to use the conditional probability of the label sequence, y, given a particular observed sequence, x, choosing the labelling that maximizes $p(y|x)$. This is the basis for the *conditional random field* (CRF) model (Wallach, 2004) employed by Shotton *et al.* This kind of undirected graphical model is frequently adopted for applications such as natural language processing, for labelling words in sentences with their corresponding part of speech tags, for example.

The conditional probability of the class labels, $\mathbf{c}$, given an input image, $\mathbf{x}$, is defined as

log $P(\mathbf{c}|\mathbf{x}, \theta)$, which is then expressed in terms of texture-layout, colour and location functions that depend on a single graph node and an edge potential function that depends on pairs of neighbouring graph nodes. The texture-layout features underlying the texture-layout potential function encode a combination of texture, spatial and textural context information and make a significant contribution to image segmentation through a Joint Boost approach to learning, which iteratively selects the best weak learner from a small subset of randomly-chosen features at each round of boosting such that each 'weak' learner is shared optimally among several object classes. The final 'strong' learner sums the confidence values of the weak learners. A pixel '*i*' is classified by evaluating texture feature responses within rectangular regions defined relative to that point. At a given pixel location, the texture feature response is the proportion of pixels that have the texture label corresponding to that feature, within the associated offset region. The system is evaluated on twenty-one object classes, but the authors believe that it could scale to cope with more, with the suggestion that in order to avoid semantic ambiguity, pixels should perhaps be able to be assigned more than one class label.

3.8 Multilevel versus shallow systems

Representing images at multiple levels enables machine vision systems to capture more complex information about scene contents. Many systems are designed on a biological basis, in which the representation at lower levels consists of 'low-level', generic features or filters that act as oriented edge detectors or texture detectors, and successive subsequent levels respond to increasingly complex structures such as corners, curves, larger parts of objects, and at the highest level, the equivalent of IT in primates, the representation can sometimes accommodates whole objects.

Marr (1982) builds a theory of hierarchical visual computation in which earlier levels extract increasingly complex 2-dimensional, viewer-centred primitives, from the raw pixel intensities of the input image to surface orientation and depth primitives of the "$2\frac{1}{2}$-D sketch", and then the highest level constructs, 3-dimensional models of objects, centred on object axes with surface shape primitives attached. The hierarchical object representation/recognition scheme of Marr

and Nishihara (1978) stores 3-D model shapes based on configurations of various sizes of cylinders, in a 'catalogue'. Various indexes allow access to different levels of complexity of the shape descriptions, and a newly-derived object description from an image is recognized on the basis of its similarity to a corresponding stored representation.

The problem of finding appropriate configurations of 3-dimensional object parts in images is a combinatorial explosion of the size of the search space, not to mention the difficulty of changing from the viewer-centred reference frame of the 2-D image to the object-centred reference frame of the 3-D object model.

3.8.1 Shallow systems

Mel's SEEMORE model (Mel, 1997) employs a shallow system that uses large set of features that collectively provides contour, texture and colour information for representing a large number of specific objects of different types, including non-rigid forms, under various views, and at different scales and locations. The features are hand-crafted, with very little spatial information included and the system is topped with a nearest neighbour classifier that takes a vector of all the feature values for a given test image as input. Thus the architecture can be thought of overall as having three levels, the individual pixel input level, the level at which pixels are organized into 'features' under relations such as being contiguous and of the same colour, or forming part of the same edge or texture region and so on, and finally, the whole object feature-vector level. However, there are some compound features, such as pairs of edges forming intensity corners that exist at an intermediate level between the basic 'feature' level and the whole object level.

Mel points out that, despite the limited amount of feature binding information within the local spatial features used by SEEMORE, objects of the same type seem to cluster together in the representation space. He relates this ability to represent global shapes with little ambiguity, based on local spatial binding with the idea of Wickelgren (Wickelgren, 1969), applied to the pronunciation of words, that, using the local context of the individual phonemes in identical unordered sets of phonemes for two words, means that the unordered sets are no longer

identical. One example given is the phonemic anagrams, /struk/ and /krust/, written in terms of context-sensitive would be represented as $/_{\#}s_{t}, {}_{s}t_{r}\ {}_{\#t}r_{u}, {}_{r}u_{k}\ {}_{u}k_{\#}/$ and $/_{\#}k_{r}, {}_{k}r_{u}\ {}_{\#r}u_{s}, {}_{u}s_{t}\ {}_{s}t_{\#}/$. Thus a global order of the words has been imposed through the use of local context.

Other shallow models operate on a similar principle of not explicitly representing global object structure, but instead, relying on often dense sampling of 'information-rich' features such as image patches. For example, Dalal and Triggs (2005), extract Histogram of Oriented Gradients (HoG) features in a dense overlapping grid, for the middle layer representation, and combine them into a feature vector for classification by a SVM.

Another example is the model of Vidal-Naquet and Ullman (2003), which has a middle layer of 'informative' image fragments. These fragments are labelled with the approximate location of where they were extracted from the image, but the relative spatial relationships among fragments are not overtly represented. Classification is on the basis of a feature vector each element of which indicates the strength of the detection of the corresponding feature in an appropriate location in the query image.

A further example of this type of three-layer representation is the 'bag-of-keypoints' model of Csurka *et al.* (2004). Again any spatial information is implicit in the feature description. The model derives a 'vocabulary' of keypoints from image patches, to form the underlying intermediate structural representation of an image, but excludes any spatial information by only representing the number of global appearances of each keypoint in the final feature vector.

Vidal-Naquet and Ullman (2003) raise the important issue that with simpler generic features, such as Gabor-wavelets, there is a need to combine them to improve their capacity to represent the characteristics of different object classes, but the difficulty is knowing which features to combine. The 'intensity corners' in Mel's (1997) SEEMORE model are an example of a simple, hand-crafted approach to the problem.

A more sophisticated approach is employed by Vidal-Naquet and Ullman (2003), using a non-linear tree-augmented Bayesian network classifier. The nodes in the tree-like architecture

represent the features and the edges indicate the statistical correlation between connected features in a probabilistic framework. The probability of a feature having a particular value is dependent on the value of its parent feature as well as on the class label. This structure allows some pairwise feature dependencies to be modelled. This is related to the notion that some features are more useful when combined with others than when used individually (Guyon, 2008; Iravani *et al.*, 2005).

3.8.2 Feature binding

This problem of what features to combine in levels of representation between the feature extraction level and the 'whole' object or image level is the 'feature binding' problem, discussed in Section 2.9.1 of the thesis. Alternatively it is referred to as the 'intermediate word' problem (Johnson, 2006). This will be discussed further in Chapter 4 of the thesis.

Another approach to explicitly binding features is the formation of 'doublet' visual words to facilitate image segmentation in the pLSA 'bag-of-words' model of Sivic *et al.* (2005). The model not only adds an intermediate topic or object representation layer between the feature and whole image layers of the architecture used by Csurka *et al.* (2004), but also pairs together words that occur within the same local region, under some predefined constraints, and uses the resulting doublets to augment the existing vocabulary.

3.8.3 Spatial information

Other systems represent even more complex relations among objects, features and parts. These are the constellations models discussed in Section 3.4.2 above. They provide a flexible model of shape and appearance variability within class. The systems of Weber *et al.* (2000) and Fergus *et al.* (2003) are fully-connected constellation models, with each part's location being modelled in relation to each of the other parts through a joint probability density. Generally the representation with such models has to be kept sparse, with only a small number of parts, usually about 3 – 7, and a limited number of features that can be assigned to the parts, say about 30, due to the computational complexity of marginalizing over all possible combinations of features being assigned to parts (Fergus *et al.*, 2003).

The combinatorial problem can be reduced by having a model in which there is less interdependency among the parts. The star model of Fergus *et al.* (2005) has a single, 'landmark' part on which all the others depend, while remaining independent of each other. This makes the processing much more efficient but since the landmark part must always be detected the model's ability to cope with occlusion is reduced in comparison with that of the fully connected model. The star representation is a tree model of depth one with the landmark part as the root node.

3.8.4 Tree-based architectures

Epshtein and Ullman (2002) employ a top-down, recursive feature extraction technique to decompose objects or parts of objects into sub-parts at successively lower levels with the process automatically terminating when no further decomposition is possible without loss of mutual information, thus forming a tree-based representation. At all levels, the corresponding features are selected on the basis of maximizing mutual information between each feature and its parent. Although the features are extracted top-down, classification occurs bottom-up, by summing the responses of all sub-features and passing the result through a sigmoid function to obtain the response of the parent feature. An overall positive response when all parent features are taken into account at the whole-object level indicates the presence of the object.

A model based on ensembles of trees is introduced by Moosmann *et al.* (2007). The argument is that using a hierarchical tree architecture, and distributing the clustering task over forests of trees to learn vocabularies of visual words is more efficient than conventional clustering techniques like *K*-means, employed in bag-of-words models such as that of Csurka *et al.* (2004). Extremely Randomized Trees employ random selection of attributes and splitting criteria to construct trees. Trees are built recursively top-down by randomly selecting a feature and a threshold and determining iteratively, using Shannon entropy, how well the split separates the classes, until the score exceeds a threshold, or a fixed number of iterations is reached. Each leaf node represents a visual word and each input descriptor is transformed by the trees into a set of

leaf indices, one from each tree. For classifying an image, a global histogram of the votes for each index is formed, as in standard bag-of-words models, and is passed to a standard classifier.

In order to cope with the difficult task of discriminating different species of insect, namely the stonefly, Martinez-Munoz *et al.* (2008) modify the model of Moosmann *et al.* (2007). The architecture of the Martinez-Munoz system has a layer of decision trees built on top of a set of random forests. The idea is to avoid constructing a vocabulary of visual words, but rather, to store at each leaf node, a histogram of the number of training instances of each class that reached the leaf during training. An important aim of this approach is to prevent the loss of information that can occur when a detected image feature is mapped to a visual word. For classification, the detected features are dropped down through all the trees and whenever a feature reaches a leaf, the 'evidence' contained in the histogram stored at that leaf is added to an overall histogram, which is then passed to the stacked classifier. Thus each feature votes for the object class, but indirectly through the evidence acquired during training.

3.8.5 Biologically-based feed-forward models

Jarrett *et al.* (2009) investigate important aspects of a hierarchical, feed-forward feature extraction and classification architecture. The hierarchy has one or multiple feature extraction levels, each one comprised of a filter bank layer, a layer that performs non-linear transformations on the filter outputs and a pooling layer that combines outputs within local regions by means of a 'max' or 'average' operation, to increase tolerance to small local variations. This basic architecture is based on what is known as the 'standard model' of feed-forward processing, theorized by Riesenhuber and Poggio (1999) to take place during the first 100 – 200 milliseconds in the primate ventral system in visual cortex, (Mutch and Lowe, 2006).

The issues addressed by Jarrett *et al.* (2009) are how the non-linearities applied after the filter banks affect classification accuracy, whether learned filters are more effective than hard-wired or randomly-selected ones, and whether it is better to have two stages of feature extraction rather than just one. The paper concludes that the most important factor in improving recognition performance is the use of a rectifying non-linearity for several possible reasons

including that neighbouring filter outputs of opposite sign might cancel each other out if being combined through average pooling. It is also found that using random filters in a two-stage architecture with appropriate non-linearities gives good recognition rates on multiple classes, and it is also found that two feature extraction stages are better than one.

The LeNet-5 convolutional neural network architecture of LeCun *et al.* (1999) incorporates the above attributes. The input to the first level is the image which is convolved with a set of planes of neural units with local connectivity that share the same weights and bias within a plane. Each plane outputs a feature map that stores the state of each unit, after adding the bias and passing the result through a squashing function, at its corresponding location. This introduces some translation invariance which is enhanced in a subsampling layer which performs local pooling by averaging over locally connected inputs and then subsampling to reduce the resolution of the feature maps. Successive alternating convolution and subsampling layers gradually increase the tolerance to distortions and relative displacements of features enabling the system to cope with the variability of hand-written digits. The architecture of the system includes a fully-connected radial basis function classifier on top.

The model of Serre *et al.* (2005) expands the 'standard model' to include four layers of processing. The S1 and C1 neural processing units of the first two layers correspond, as in the model of Riesenhuber and Poggio (1999), to the simple and complex cells, respectively, in the primary visual cortex, V1, with the S2 units of the next level learning a local prototype representation and the fourth level C2 units showing similar selectivity and invariance to stimuli as in primate IT cortex.

The input image is first processed by the S1 Gabor filters of different orientations and scales. Then the S1 outputs are subsampled by C1 units that pool over afferents from a local neighbourhood, taking the maximum over location and scale at a given orientation. At the next stage, S2 units each compute the similarity, in C1 format, between image patches at all locations and a learned prototype patch for a preferred orientation and scale. Finally, max-pooling as

applied to the S2 units associated with a particular patch to produce a vector of length equal to the number of prototype patches, Serre *et al.* (2006).

Serre *et al.* (2006) argue for the lack of spatial information included in this model compared, for example, with constellation systems, on the basis that biological vision is unlikely to be able to make use of it in the ventral stream, at least., indicating that the performance of the standard model is comparable to the non-biological approaches.

There are a number of problems with the standard model, including the high computational cost of matching an image against a dense representation of prototype features and the likelihood that such a dense representation will be noisy, leading to misclassifications.

Also, the maximum pooling operation discards potentially useful information from the neighbours of the unit with the strongest response and in addition, the system has no method of feature selection to obtain a manageable set of reliable features (Huang *et al.*, 2008).

3.8.6 Modifications of the standard model

The model of Serre *et al.* (2005) has been further developed by Mutch and Lowe (2006). The biologically- motivated modifications in Mutch and Lowe's version apply sparsification to the S2 representation and apply lateral inhibition in the S1 and C1 layers to reduce noise and improve generalization performance. In addition, spatial information is incorporated on the basis that neurons in V4 and IT are not fully invariant to location and scale and their receptive fields do not necessarily cover the whole visual field. This is modelled by requiring that an S2 feature be matched within a limited region centred on where it was detected in the image of origin. Also a feature selection stage is added to improve on the randomly-selected S2 features by eliminating less 'useful' features, by dropping features that are assigned low weights by an SVM classifier in an iterative process.

The 'enhanced biologically inspired model' of Huang *et al.* (2008) also seeks to make the representation more sparse. This is achieved by only extracting features from regions of interest as measured by gradient. It also adapts the max pooling operation between simple and complex

units by summing the energy of the maximum response and its neighbours and rejecting the remaining weak responses. This is based on a contrasting view to that of Mutch and Lowe (2006), that the maximum response suppresses the activity of its neighbours, which principle they employ to increase sparsification of S1 and C1 outputs. Conflicting theories of sparseness are discussed in Section 2.8 of the thesis.

Feature selection in the Huang *et al.* (2008) model is achieved through feedback in the form of a cascade of feature-rejecters using AdaBoost in a way similar to Viola and Jones (2001).

3.8.7 Perception in multilevel systems

Wolf *et al.* (2006) explores some types of architecture in terms of which levels are involved in merely building a representation and which are also involved in perception and which levels contribute to the final perception decision at whatever level that occurs. The work is inspired the Reverse Hierarchy Theory of Ahissar and Hochstein (2002), introduced in Section 2.8 of the thesis, which, as explained by Wolf *et al.* (2006), posits that while visual information initially travels bottom-up through the feed-forward hierarchy, perception starts at the higher levels with general information about the gist of a scene, and travels down the hierarchy through feedback connections as more detailed information is required. The authors experiment with five different strategies for classifying in hierarchical systems, Figure 3.9 (from Wolf *et al.*, 2006, Figure 2).

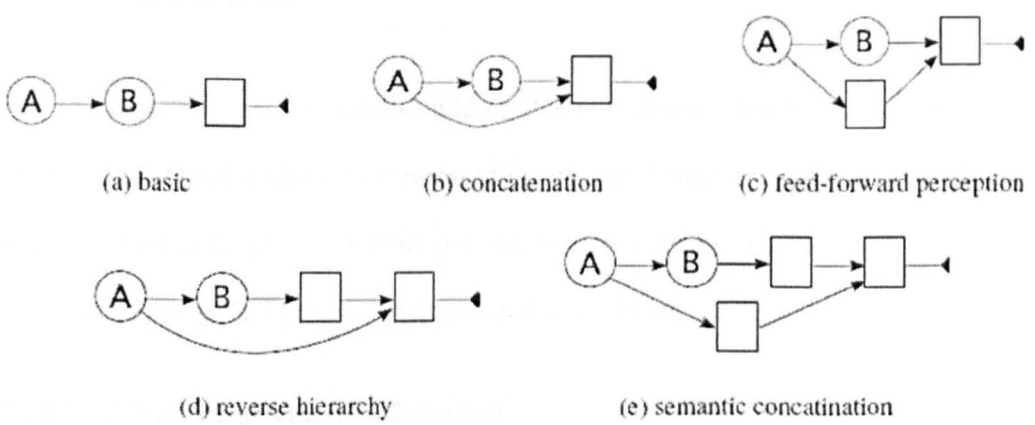


Figure 3.12: Some perception hierarchy architectures
The circles indicate representation and the squares represent classification
from Wolf *et al.*, 2006, Figure 2

In the basic model, classification is usually carried out at the highest level, as in Serre *et al.* (2005). The concatenation strategy combines all the features from all levels into a single vector for classification. This is the approach of Bileschi and Wolf (2006), combining the Gestalt descriptors with the more basic features. The feed-forward perception strategy, reverse hierarchy and semantic concatenation models all require more than one classification layer.

The reverse hierarchy bases the initial perception on the high-level 'gist' of the scene. This perception is then classified along with the low-level features. This approach is put forward by Oliva and Torralba (2001) as a way of using the context as a first stage in object detection by priming objects that are more likely to occur.

The stacked evidence trees model employed by Marinez-Munoz *et al.* (2009) for discriminating very similar objects is an example of the semantic concatenation approach. In the model, the evidence is gathered at the first 'classification' level is stored in the form of histograms. Then the descriptors for a new input image are put through the forest and the histograms at the receiving leaf nodes are updated and the histograms at each node are summed to form a single overall class histogram which is fed as a vector to a boosted ensemble of decision trees at the next level. The overall finding is that feedback does seem to help in object recognition.

3.9 Conclusions

The ideal machine vision system should be able to represent, detect and classify many different categories of object under a wide range of image transformations, such as changes in location, scale, rotation in the plane, rotation in depth, as well as being able to cope with partial occlusion, changes in lighting conditions and noisy input.

3.9.1 What has been achieved

The review in this Chapter has shown that many systems are tackling these issues quite successfully and there is a move away from models that are highly designed by the user. It is

now unusual to find a system that extracts features considered by the designer to be useful, for example, eyes, nose and mouth features for face recognition. Instead, features are often densely sampled from images and then a 'useful' subset is selected. This more flexible approach to finding features extends the recognition capacity of a system to a larger range of objects. Nevertheless, it is difficult to find a descriptor that is appropriate for all circumstances. For example, texture-based features such as SIFT (Lowe, 1999) are not likely to be as effective as contour-based ones in images in which the outlines of objects are the main source of information.

One way to tackle this problem might be to choose a large assortment of many different types of features as in Mel's SEEMORE model (Mel, 1997). However, there is the problem that features that are not well-suited to the task in hand can adversely affect performance. Selecting the best features from among a large initial set is an important factor and there are issues to consider such as the relevance of features, their mutual dependence and the optimal number of features.

The review has discovered that various techniques are employed successfully to address different aspects of the problem, for example, PCA and LDA for dimensionality reduction (Martinez and Kak, 2001), clustering for construction of 'prototype' features (Jurie and Triggs, 2004), selecting relevant subsets of features (Ullman and Sali, 2000; Kira and Rendall, 1992), and eliminating redundant features (Bart and Ullman, 2004; Weber *et al.*, 2000).

The quality of feature selection can also contribute significantly to image segmentation and there is a variety of approaches ranging from explicitly segmenting an image before performing object recognition, which can be effective when domain knowledge is available in a particular task (Johnson and Simon, 2001), to segmenting as part of object detection to overcome the problem of different objects sharing similar parts (Shotton *et al.*, 2009).

There are many different architectures to choose from, ranging from shallow systems, such as 'bag-of-words' models that do not include any spatial information (for example, Dalal and Triggs, 2005; Csurka *et al.*, 2004), to fully-connected generative constellation models that represent the relative spatial distributions of all the constituent object parts (Fergus *et al.*, 2003;

Fei-Fei *et al.*, 2007), to biologically-inspired multilevel systems (Serre *et al.*, 2005; Jarrett *et al.*, 2009), in which spatial information is implicit in the local connectivity of feed-forward processing at successive representation levels.

Building a multi-class classifier can be a problem. Csurka *et al.* (2004) employ multi-class comparison of seven classes of object, however, breaking a multiple class classification task into several binary ones is often considered simpler than using a single classifier for all classes. Also, classifiers like SVMs are ideal for solving binary problems. There are two main approaches to dividing a multi-class classification task, one-v-all, in which each classifier is trained to discriminate one class from all the others, and pair-wise classification, where each classifier learns to discriminate a pair of classes. The former scheme requires one classifier for each class, while the pair-wise system needs $K(K-1)/2$ classifiers for K classes (Zhou *et al.*, 2008). Apart from requiring more classifiers, the problem with the latter approach is to decide how best to group the classes for each classifier. Zhou *et al.* (2008) adopts an Error Correcting Output Coding approach to optimize the number of classifiers and the classes they each learn to discriminate, using information about the separability of the various classes and the distribution of the data within them.

Also it is important to consider carefully how progress is evaluated in designing such systems. A somewhat controversial view on the claims of success of some state-of-the-art systems in classification tasks using multiple class data-sets of 'natural' objects, is aired by Pinto *et al.* (2008). The paper urges caution with respect to the belief that data-sets such as Caltech 101 (Fei-Fei *et al.*, 2003) are a real test of an artificial visual system's ability to cope with image variations in location, size, orientation, lighting and so on. The argument is made on the basis that in these data-sets, typically image transformations are not varied systematically, image backgrounds often co-vary with object class and many images are composed by the designer and so are not necessarily very representative of real-world scenes. An experiment testing a baseline classifier modelled on low-level primate simple cell responses in area V1 against five state-of-the-art systems, including that of Mutch and Lowe (2006) shows that the baseline classifier performs at least as well as the more sophisticated systems. The baseline classifier is

then tested on what should be a simpler 2-class discrimination task devised by the authors, in which performance deteriorated dramatically as variations in pose scale and location were steadily increased. The conclusion is that the 'natural' image sets currently being used as a benchmark for test object recognition systems do not contain enough of the complexity of real-world scenes. Suggestions to remedy this include the generation 'natural' image databases without any bias in how the images are captured, or the use of synthetic images to portray the full range of image transformations of objects in the real world. The importance of using baseline classifiers to help in judging the difficulty of different visual tasks is also emphasized.

3.9.2 What has still to be achieved

It is not only that a system should be able to detect or categorize a large number of different types of object, but that it can learn new classes without having to retrain from scratch. Some progress has been made in this area, for example, in the work of Fei-Fei *et al.* (2003) and Bart and Ullman (2005).

What is needed is machine vision systems that can adapt to any new visual challenge, that can extract information from images in a form appropriate to the task and can modify their architecture in order to build the required level of representation complexity.

With the considerable variety in the design of multilevel systems, it is difficult for the user to decide on the best design of system for a particular visual task. What is lacking is a common set of principles for constructing such a system, whether or not the preference is for a biologically-based representation, or one based on more explicit configurations of object parts, for example.

In addition, setting the parameters for the connectivities within and between levels in such a system is a problem that may be better addressed in a task-specific way, rather than with a fixed 'one-size-fits-all' approach, as, for instance, in the standard biological model of Serre *et al.* (2005). The ideal situation would be for a system to be able to design its own connectivities in response to current requirements and the information extracted from the input data and to be able to adapt to changing circumstances.

Although progress has been made in the area of managing the often large quantities of data generated in the process of feature extraction through a variety of feature selection techniques, these methods are generally applied at a single representation level, usually the level at which classification is to be effected, and with the exception of systems like that of Epshtein and Ullman (2005), no attempt is made to select features at multiple levels of complexity, nor to learn the optimal number of representation levels for a task. Ideally a system would 'discover' such features and the optimal number of levels through knowledge of the connectivities established during feature extraction.

The above observations have lead to the formulation of the research questions listed below.

3.9.3 The research questions

- Is there a general architecture for representing multilevel systems, the same 'formula' being appropriate for a wide variety of representation/recognition problems?
- Can such systems be self-forming?
- How can systems find their own descriptors?
- Is there a way that structure at higher levels can 'emerge' so that the intermediate word problem and the combinatorial and dimensionality problems can be solved automatically?

The methodology applied to the exploration of these questions is explained in Chapter 4.

Chapter 4: Towards autonomous feature selection and adaptable architectures for object recognition

4.1 Introduction

Chapter 2 has reviewed theories of how biological vision systems, in particular primate vision, tackles the main problems inherent in object representation and recognition. In Chapter 3, current research into artificial object recognition systems was discussed, and it was found that despite the incorporation of biological vision concepts into the design of artificial systems, there are still considerable difficulties to be overcome in specific areas for machine vision systems to be able to function autonomously, learning their own representations and adapting to new tasks.

In particular, the areas that were identified were:

Feature extraction – Systems need to extract features in such a way that the process automatically adapts to find appropriate features for different visual tasks.

Feature selection – In selecting relevant features, systems need to be able to determine which features are able to 'see' objects belonging to the same class as being 'similar' in some way, and objects from different classes as being 'different'. The way 'similarity' is measured is important.

Representation architecture - Systems need to be able to form representations at multiple levels of complexity and automatically adapt their architecture in response different visual tasks and user requirements, to enable an appropriate level of abstraction to emerge for successful classification.

Chapter 4 introduces the algorithms that are tested in the experiments described in Chapter 5. Section 4.2 presents two different approaches to enabling systems to extract features autonomously from images. In Section 4.3, four different ways of selecting 'useful' subsets of features are presented. Section 4.4 discusses the problem of measuring similarity between objects and explains how *hypernetworks* can enable similarities to be represented explicitly. In Section 4.5, the concept of using *hypernetworks* for representing multilevel structure and as a

means to enable systems to classify at different representational levels, is introduced. Section 4.6 provides the summary.

4.2 Autonomous feature extraction

This thesis investigates the problem of autonomous feature extraction through two different approaches:

- Random generation of simple, minimally-constrained pixel-configurations
- Algorithmic generation of homogeneous and heterogeneous polygons

4.2.1 Random feature extraction

As noted in Chapter 3, many feature-types are highly engineered to be useful for extracting information from particular kinds of image. Some are useful for images that contain a lot of lines, while others are better suited to describing different textures.

Primitive vision systems can detect light and dark regions on the retina, so the principle behind the random feature extraction approach was to generate the simplest type of feature that would enable the system to detect differences in intensity at different points in binary images, the aim being to be able to discriminate simple geometric shapes such as circles, diamonds and squares. Details about the data used are given in Chapter 5, Section 5.2.1. A feature consisted of a pair of pixels the locations of which were randomly chosen, with no constraint on orientation or distance, to fully explore the effectiveness of a global approach to describing and classifying simple geometric shapes using such basic constructs. The idea was that this random, minimally-constrained approach to feature extraction could enable systems to adapt their representation as required (Rose and Johnson, 2005).

In a non-randomly derived Bayesian network-based, face recognition model, non-local pixel-pair configurations have been found to be useful in non-binary images for detecting the relatively unchanging 'faces' of objects against 'cluttered' backgrounds (Pham and Smeulders, 2006). Similarity in greyscale between nearby pixels is commonplace, but strong correlation

between distant pixels occurs much less frequently, giving a good indication of the presence of objects such as faces.

In binary images, there are four possible 'light-dark' patterns of pixel pairs, Figure 4.1. Two of the configurations indicate no change between the two pixels, implying to the system that there are no edges between the two locations, which may or may not be true, as a comparison of the patterns labelled '0' and '3' in the figure illustrate. Patterns '1' and '2' show there has been a change so that there must be at least one edge between the selected points.

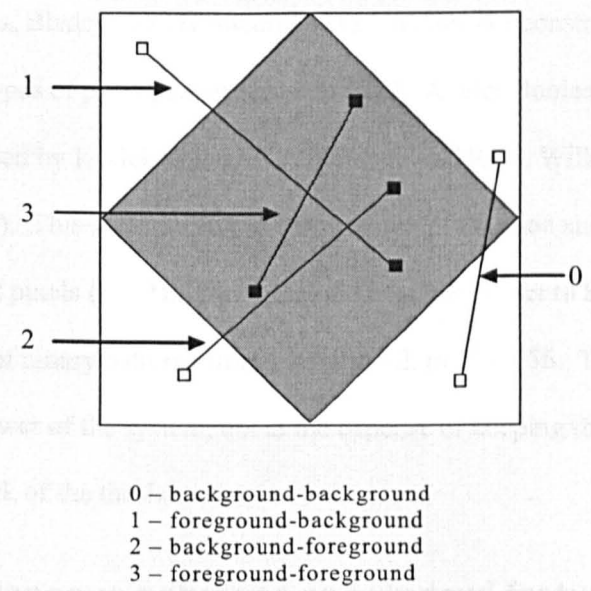


Figure 4.1: The four pixel-pair configurations

The number of pixel-pair features extracted was varied to discover how dense the sampling would have to be to discriminate the shapes for which there is considerable overlap of foreground pixels and hence much redundancy likely to occur in the representation.

The effect of restricting the permitted degree of variation in the spatial relationship between pixel-pair members was also investigated to determine whether more local representation would perhaps improve the reliability of 'edge' information. Also the effect of limiting the types of configuration to be extracted was tested to see if some combinations of patterns were more useful for classification than others. Relative 'usefulness' was judged on the basis of

classification accuracy and on whether there was a change in the number of pixel-pairs required for the same classification performance.

This random pixel-pair feature extraction process is similar to that of the Boolean Neural Network system of W. W. Bledsoe and I. Browning (1959), described in Picton, 1994 (p46). (Boolean networks are composed of combinations of Boolean logic elements, such as the 'AND'-gate, and require binary input.) However, Bledsoe and Browning's system randomly pairs off *all* the pixels in the input image, whereas, in the work of the thesis, only a subset of the possible pairings is extracted, and, in addition, the number of pairings can be varied. Also, unlike in the thesis, Bledsoe and Browning's system does not constrain spatial relationships, nor does it limit the types of pixel-pair pattern extracted. A later Boolean Network system called 'WISARD', devised by I. Aleksander, T. J. Stonham and R. A. Wilkie (1982) is described in Picton, 1994 (p49). This system extends the program of Bledsoe and Browning to randomly extract n-tuples of pixels ($n > 2$). The value of n is generally set to 8 which increases the possible number of binary patterns from 2^2, for n = 2, to $2^8 = 256$. This enhances the discriminatory power of the system, but at the expense of keeping the features simple, as was the aim in the work of the thesis.

4.2.2 Homogeneous polygons as 'window' feature descriptors

While the pixel-pairs features were sampled relatively sparsely, the approach with the 'window' features described in this section was to sample the images densely and then select a subset of the most 'useful' windows for object discrimination. This process of feature selection and the motivation behind it is discussed in section 4.3.3. The focus here is on how the window features were encoded using polygons as descriptors. The data that were processed using this approach were low-resolution greyscale images of pedestrians in street scenes and general outdoor scenes with no pedestrians, from the DaimlerChrysler Benchmark Data Set (Munder and Gavrila, 2006) for use in a pedestrian recognition task, examples shown in Chapter 5, Figure 5.37, referred to hereafter as the 'NiSIS' dataset (from the NiSIS pedestrian recognition competition, 2007).

The idea was to take local, overlapping rectangular 'window' samples and in each window, establish the local average greyscale value and assign pixels with the average or greater greyscale the label 'light' and pixels of lower than average greyscale the label 'dark'. Contiguous pixels with the same label were then assembled to form polygons as explained in Chapter 5, Section 5.3.1.1, and the resulting 'light' and 'dark' polygons were described in terms of a set of simple attributes. For each polygon, the following measurements were obtained: the greyscale variance about the polygon's mean value; the variance in both the 'x'- and 'y'-directions around the polygon's centre of mass; the direction of the centre of mass of the polygon to the centre of the window in which it occurred.

This approach of dense sampling to produce a redundant, overcomplete representation, is similar to that of Papageorgiou *et al.* (1998), in a pedestrian recognition task, except that in that work, the window features are based on Haar wavelets, rather than their polygonal content. Maree *et al.* (2005) also use window-based features, encoding them in terms of their raw colour pixel values, but the sampling is random rather than exhaustive, in a multiple object-category recognition task. The appeal of dense sampling is that it potentially provides a richer description of the input than approaches where sampling is more sparse, often being limited to small neighbourhoods around interest points. In addition, interest points tend to be defined by the system designer, whereas the aim in this work is to enable the feature extraction process to be as autonomous as possible.

As well as the variability in the shape and size of the polygons, there is also the consideration that the number of light and dark polygons in each window varies, since polygon formation is sensitive to changes in greyscale value – even a change in the value for a single pixel could cause polygons to merge or split. Therefore a window could have a different length of description vector for representing the same location in a pair of pedestrian images that look similar, and the variability in the windows representing similar positions in the non-pedestrian images would be likely to be even greater due to the considerable variation within that class. To keep the comparison between pairs of correspondingly-located windows simple, it was decided

to only allow comparison where the windows in question had the same length of vector description and furthermore, matching numbers of light and dark polygons. However, it was realized that, due to the problem that regions of similar appearance could be represented by different numbers of polygons, there was the risk that opting for simpler comparison might cause the system to 'miss' some close similarities during feature selection and image classification.

4.2.3 Non-homogeneous polygons as features

The previous feature extraction approach extracted user-designed 'window'-features and encoded them using descriptors that the system could extract autonomously simply by taking an average greyscale within the window and forming polygons of contiguous 'like' pixels, 'like' being defined very loosely as being on the same side of the average, thus segmenting the window content into homogeneous light and dark regions.

The aim was now to increase the autonomy of the feature extraction process still further and explore whether automatically generated, non-homogeneous sets of pixels, comprising a mix of light and dark pixels, could provide enough information about the structure of objects of different categories for reliable discrimination. The extraction process made use of a simple 'region-growing' algorithm previously employed very successfully in aiding biological research by locating cells in images of developing organisms by defining their boundaries, Johnson, (In Press).

Most region-growing algorithms work on the basis of gradually adding more and more points or pixels that are similar and within a certain preset proximity to those already included in the current region. The difference with this approach is that a new member pixel is added to the region or polygon if it is the most different to the most recently added pixel. This enables the system to form a set of polygons that roughly mark the edges of an object and thus effectively segment an input image, provided the background is uncluttered.

The algorithm was applied to examples from the MNIST database of hand-written numerals (Lecun and Cortes, 2010) the images of which are effectively binary. The polygon formation procedure can begin anywhere in the image, but in this work is initiated with the top leftmost pixel starting by searching its available 8-neighbours for the pixel that differs from it the most in greyscale, the difference being required to be at least four greyscales. If no pixel is sufficiently different, the algorithm proceeds to the next pixel and examines its 8-neighbours. Once a sufficiently different 8-neighbour that differs most from the centre pixel is found, the formation of a polygon begins. The next step is to examine the 8-neighbours of the newly added pixel and again select the most different of its 8-neighbours, above a threshold, to add to the polygon. If in the formation of a polygon there are no neighbours that are sufficiently different, or if all the neighbours have already been assigned to other polygons, the process for that polygon is finished, and the next unassigned pixel in the list is then used as the start of a new polygon. Thus no pixel can belong to more than one polygon. The algorithm terminates when all the pixels in the image have been visited. Table 4.1 shows an example of how polygons are generated. It contains an extract of greyscales that comprise the upper left region of a '0' from the database.

	a	b	c	d	e	f	g	h	i
A	255	255	239	95	19	2	2	2	1
B	255	255	156	2	2	2	2	2	1
C	255	230	61	2	2	2	2	124	158
D	255	49	2	2	4	22	128	246	255
E	200	15	2	2	22	255	255	255	255
F	79	2	2	2	128	255	255	255	255
G	79	2	2	124	246	255	255	255	255
H	1	1	23	180	255	255	255	255	255
I	2	2	101	255	255	255	255	255	255

Table 4.1: Polygon generation: The 'growing sequences' of three polygons (cells are pixel greyscale)
first polygon: Ab → Bc → Bd → Ac → Ad → Be → Ae → Af
second polygon: Ba → Cb → Dc → Cc → Bb
third polygon: Bg → Ch → Di → Ci → Bi

The importance of this approach is that the resulting polygonal constructs are not engineered by the system user, but instead are the result of the system applying simple instructions, based on a kind of meta-knowledge of which it is unaware, that in an uncluttered background the algorithm

will automatically find and envelope objects of interest with these constructs. Figure 4.2 shows examples of the sets of polygons generated by some of the numerals. These sets are referred to as the *polygon envelope* of the numeral.

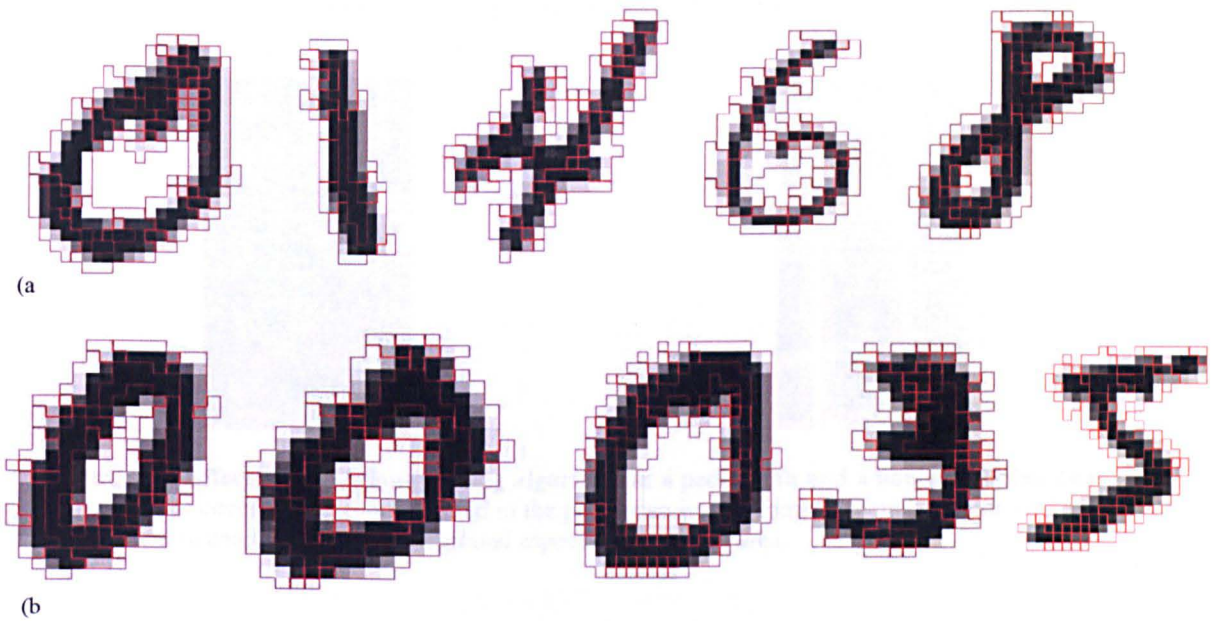


Figure 4.2: Numerals from the MNIST database with their 'polygon envelope'
(a) They are located easily against a plain background.
(b) Illustration of the difference in number, size and distribution of polygons for within class exemplars (the three 0s), and of the potential for different classes of numeral to have a similar number and size of polygons

However there are some problems. Although the algorithm generates a fairly compact envelope, the output is unstable in a similar way to that of the algorithm for generating the homogeneous polygons described earlier, in the sense that a small change in greyscale in an image can give rise to considerable changes in the polygon envelope. Thus the polygons generated vary considerably in size and shape and the number and configuration of polygons in the envelope is also very variable, not only among different classes of numeral but within class as well, Figure 4.2(b), making it a challenge to achieve consistency in the image descriptions for each class of numeral.

The algorithm was also applied to pedestrian and non-pedestrian greyscale images, in the pedestrian recognition task that was initially tackled using window features described by the homogeneous polygons.

The major problem with generating the heterogeneous polygons in greyscale images like those in the NiSIS database is that there is no longer an uncluttered background to allow a relatively 'clean' segmentation of the object from the surroundings. Instead, the algorithm indiscriminately grows polygons that completely cover the images, often spanning relatively large portions of background as well as foreground in the pedestrian images, Figure 4.3.

Figure 4.3: Effect of the region-growing algorithm in a pedestrian and a non-pedestrian image
The polygons cover the entire image, and in the pedestrian image, some of them span extended regions of background in conjunction with foreground especially in the leg area.

For both the hand-written numerals and the pedestrian recognition data, applying the algorithm to an image generates a large number of polygons, many of which are likely to be quite uninformative due to factors such as their location in the image and their size and shape. Therefore as with the window-features that were used initially with the pedestrian data, feature selection would be required to help identify a subset of potentially useful 'classifier' polygons, as described in Section 4.3.

As with the homogeneous polygons, the heterogeneous constructs were encoded using a set of simple descriptors. Measurements of greyscale variance would have been less appropriate for these polygons, and so descriptors in the form of sixteen 2x2 pixel patterns were chosen, Figure 4.4.

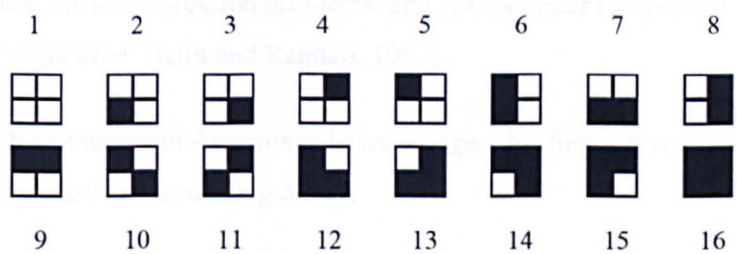


Figure 4.4: The sixteen 2x2 pixel patterns from Rzevski (Ed), 1995, p114.

Polygons were binarized using the local greyscale average as described earlier, and then encoded using the above patterns to form a 16-dimensional vector of the number of occurrences of each pattern comprising the polygon. Details of the encoding process are given in Chapter 5.

4.3 Feature selection

As discussed in Chapter 3, Section 3.3, high-dimensional representation can make it difficult for machine vision systems to learn to classify patterns, especially when relatively few training examples are available, due to the problem known as the 'curse of dimensionality', in which patterns are sparsely distributed and do not readily form distinct, compact clusters corresponding to different classes in the representation space. In addition, many of the dimensions tend to be irrelevant, outputting a similar value across multiple different classes of pattern, thus causing patterns of different classes to appear more similar than they actually are. Also, having too many features can lead to a reduction in the ability to generalize to new data, especially if there are correspondingly too few training examples. Therefore finding ways of selecting an appropriate subset of relevant features is of great importance, and a number of different methods were discussed in Chapter 3.

In this thesis, feature selection is addressed in four different ways:

- Iteratively randomly generating sets of features and selecting the best-performing set.
- Constraining the representation to include only a particular type of feature.
- Using a modified form of an existing feature-ranking algorithm, introduced in Chapter 3, that has many variants in the literature for application in different types of classification and regression problems, and comes under the general heading of the '*Relief Algorithm*' (Kira and Rendall, 1992).
- Through an incremental approach based on 'greedy' forward-search applied to the features initially ranked using *Relief*.

4.3.1 Feature selection by choosing the best classifier set from a number of randomly-generated sets

To form the representation of a simple shape, the randomly-generated pixel-pairs described in Section 4.2.1 were generated iteratively to form sets of patterns. A shape was then represented by an *n*-dimensional vector of pattern numbers, (0, 1, 2, 3) Figure 4.1, where *n* is the number of pixel pairs in an individual set. A wrapper approach was then applied to selecting the best feature set, by using an 'evaluation' set of shapes for comparing their classification performance.

4.3.2 Feature selection by restricting the representation space

As shown in Figure 4.3(b), with the hand-written numerals, the polygon envelope produced by applying the non-homogeneous polygon generating algorithm varies considerably within-class and so some way was required to try to create a more consistent intra-class representation, while making inter-class differences more distinct. The approach taken was to try to standardize the individual polygon descriptions so that polygons that were similar in some way could be represented as belonging to the same set. Three different generalization strategies were explored and tested on the '0' and '1' numeral classes:

Polygons of one particular size were considered, in this case, polygons containing just two 2x2 patterns, referred to as 'size-2 polygons. The actual 2x2 configurations were not taken into account. Training examples were divided into subsets according to the number of size-2 polygons they contained, and test items were also categorized on the basis of how many size-2 polygons they contained, being assigned to the class with the larger quantity of training examples containing that number of size-2 polygons.

Polygons of all sizes, in terms of their constituent 2x2 patterns, were involved, but rather than considering each size separately, the polygons were assigned to three categories: – small, medium and large. To reduce the dimensionality of the problem still further, the sixteen 2x2s patterns were reduced to just three: – light, medium and dark as shown in Figure 4.6, and

initially only the light and dark sets of patterns were used. Now, a polygon was categorized according to whether it was small, medium or large and whether the number of constituent light 2x2s was greater or less than the dark 2x2s, and each numeral was represented in terms of the number of polygons of each type comprising its polygon envelope. During classification, only images with envelopes containing the same number of polygons were compared.

Again, polygons were described as small, medium or large, but all three types of 2x2s pattern, light, medium and dark, were taken into consideration for this representation, with a polygon being categorized according to size and whether, for example, the number of light 2x2s exceeded the number of medium and dark 2x2s. As before, numerals were classified on the basis of the number of polygons of each category in the envelope.

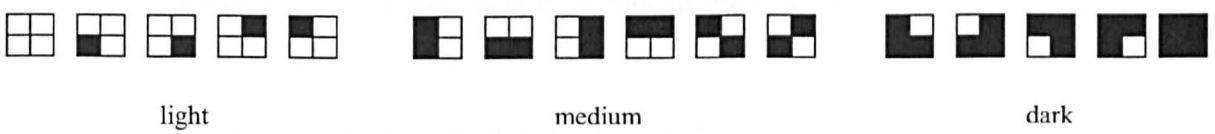


Figure 4.5: 2x2s patterns assigned to just three different categories - light, medium and dark

These three approaches are explained in more detail in Chapter 5, Section 5.7.

The 0s and 1s were not very well recognized using the above generalization methods. It is likely that too much important information about the polygons was being discarded and that possibly the restriction to only matching polygon envelopes of the same size was too limiting.

Another major problem was that simply counting the frequency of occurrence of various types of polygon meant that important spatial information about the relative positions of polygons of various types within the envelope was being omitted. So a new approach was needed. Rather than trying to characterize each numeral by encoding its polygon envelope through a 'standardized' representation of the constituent polygons, the system would generate, using a subset of training images of different classes, a pool of polygons from which some could be chosen, according to predetermined criteria, to provide a fixed set of 'polygon windows'

through which to inspect incoming images. The potential advantages of using a pre-selected set of polygons are:

- A fixed-size set of polygons can be employed, which does away with the necessity to try to standardize the numeral envelope, or try to compare objects with variable-length descriptions. The polygon envelope can instead be used as a means of locating an object of interest in an image, roughly segmenting it out from the background – currently only in the context of a plain background.

- The absolute location of a polygon in relation to the image frame and its position relative to the centre of the object's polygon envelope can be known, rather than either having the problem of trying to determine the relative position of polygons within a very variable envelope or, abandoning spatial knowledge in an attempt to reduce the complexity of the representation, as in the initial approach.

- A repertoire of polygon windows can be built up gradually as the system is required to learn new objects.

- Multilevel representation (introduced in Section 4.5) is simpler because the relative locations of the polygons are known, making the formation of higher-level structures more consistent.

The incremental building up of a repertoire of polygons as more object classes are introduced is described in Section 4.3.4.

A possible disadvantage of this approach is:

- Each polygon that is generated is part of the envelope of one particular image and will therefore not generally 'fit' into the envelope of another image even within the same class, so there is a question about what such a polygon is actually going to detect. The structure being detected by a polygon window might be rather variable in that the polygon might 'miss the target', perhaps tending be positioned sub-optimally leading to the inclusion of too much background at the expense of the foreground.

4.3.3 Feature selection with 'Relief'

The pseudo code for the basic Relief algorithm, given in Chapter 3, is reproduced in Figure 4.6.

Algorithm Relief
Input: for each training instance a vector of attribute values and the class value
Output: the vector W of estimations of the qualities of attributes

1. set all weights $W[A] := 0.0$;
2. **for** $i = 1$ **to** m **do begin**
3. randomly select an instance R_i;
4. find nearest hit H and nearest miss M;
5. **for** $A := 1$ **to** a **do**
6. $W[A] := W[A] - \text{diff}(A, R_i, H)/m + \text{diff}(A, R_i, M)/m$;
7. **end**;

Figure 4.6: The basic Relief algorithm
from Robnik-Sikonja and Kononenko, 2003, Figure 1.

The principle of the Relief algorithm is that good attributes or features should be able to distinguish between exemplars that are close to each other and should be 'rewarded' according to how successful they are. For each of m randomly selected instances, R_i, the algorithm searches for its two nearest neighbours, one from the same class, referred to as the nearest hit H, and the other from the other class, termed the nearest miss M – line 4 above. It increases the value of an attribute, A's, weight if there is a smaller difference in the value of A when comparing R_i with H than when comparing R_i with M – line 6. This difference is obtained using the diff() function of line 6 given below as equation (4.1):

diff(A, I_1, I_2) calculates the difference in the values of an attribute for two instances I_1 and I_2.

$$\text{diff}(A, I_1, I_2) = \frac{|\, value(A, I_1) - value(A, I_2) \,|}{max(A) - min(A)} \qquad (4.1)$$

The diff() function in equation (4.1) is also used in finding the nearest neighbours, with the overall difference between two instances taken as the sum of all the differences between the individual attributes. Polygons generated by the MNIST data were compared on the basis of the overall difference in greyscale of correspondingly located pixels within the polygon window, ie pixel-matching. Exemplar A is considered more similar to neighbour B than neighbour C if the sum of the differences in greyscales between A and B is smaller than that between A and C.

In the pedestrian data the polygons were compared on the basis of the frequency of occurrence of each of the sixteen 2x2s patterns. Exemplar A is considered more similar to neighbour B than neighbour C if the sum of the differences in the number of 2x2s pattern counts between A and B is smaller than that between A and C.

The Relief algorithm can cope with nominal features where the diff function is a sigmum function that takes the value 0 if the features being compared have the same value and 0 otherwise. The features in this work are numerical and their values are compared using equation (4.1).

In this work, the algorithm has been adapted so that the description of each window or heterogeneous polygon is treated as a 'compound' feature or attribute of the image to which it belongs and the difference between two such attributes is the sum of the differences between the individual elements in the vectors representing them. In the case of the window features described in Section 4.1.2, this sum of differences is divided by the dimension of the vectors, to ensure that longer vectors do not tend create larger overall differences.

Also, to avoid the problem of high dimensionality, due to the large number of windows (208) and heterogeneous polygons (over a thousand), the difference between the randomly selected image instance and each nearest neighbour is calculated using only the window or polygon currently being evaluated. This is at the expense of being able to consider the contribution to 'closeness' of all the features when selecting the nearest neighbours, Guyon, 2008, Section 2.4, but it has the benefit of ensuring that the chosen nearest neighbours are close on that particular feature.

As indicated in the pseudo-code of Figure 4.7, the basic *Relief* algorithm can only be applied to 2-class problems. The pedestrian recognition problem is, of course, a 2-class problem and in this work, the hand-written numeral task, although a multi-task problem, only extracts heterogeneous polygons from two numeral classes, applying *Relief* in a 2-class context, and then

using the resulting pool of relevant polygons to provide potentially useful discriminators for the other classes as well. This process is explained in more detail in Chapter 5.

However, despite the applicability of the original Relief algorithm in this work, it was felt that, due to the high variability in the both the pedestrian and numerals data, a more robust form of the algorithm based on *Relief*F (Kononenko, 1994) that is not limited to 2-class problems and is better able to cope with noisy data, was required.

The full version of *Relief*F is given in Robnik-Sikonja and Kononenko (2003, Figure2), and the extended version of *Relief*, used in this work, modelled on the *k*-nearest neighbours approach of *Relief*F is shown in Figure 4.7 below. The main difference to the basic Relief is that, instead of just one example of each class, *k* nearest hits and misses, $k >= 2$ are selected on each iteration of the algorithm, Line 4.

Input: for each training instance a vector of attribute values and the class value
Output: the vector W of estimations of the qualities of attributes

1. set all weights $W[A] := 0.0$;
2. **for** $i = 1$ **to** m **do begin**
3. randomly select an instance R_i;
4. find k nearest hits H_j and k nearest misses M_j;
5. **for** $A := 1$ **to** a **do**
6. $W[A] := W[A] - \Sigma_{j=1}^{k} \text{diff}(A, R_i, H_j)/(m.k) + \Sigma_{j=1}^{k} \text{diff}(A, R_i, M_j)/(m.k)$;
7. **end**;

Figure 4.7: The modified Relief algorithm used in this work
adapted from Robnik-Sikonja and Kononenko, 2003, Figure 2

Line 6 in Figure 4.7 indicates that the weight for each feature is updated using the average difference over the k hits and the *k* misses as well as the number of iterations, *m*, of the algorithm.

Application of the algorithm produces a ranked set of features, some of which have very low, or even negative weighting. A subset of the best-scoring ones can then be selected by applying a threshold. In this work, the threshold was determined as the average of the non-negative scores, and features with a higher than average weighting were selected, either to form the

representation of the pedestrian and non-pedestrian images, in the case of the 'window' features, or as a pool ready for a further selection process in the case of the heterogeneous polygons.

In the pedestrian recognition work involving the window features, the training images were also ranked according to how frequently they were involved in a correct matching during the application of the Relief algorithm, Chapter 5, Section 5.6.3.

4.3.4 Incremental feature selection for learning new object classes

In primate vision, it is not generally necessary to see numerous examples of a new object to learn to recognize it. Often a single instance is sufficient, for example, learning a new face can be accomplished rapidly, with a single showing (Rolls and Deco, 2002, p120).

The idea of not having to introduce large amounts of training data for every new class has attracted considerable interest in machine vision research, as discussed in Chapter 3, Section 3.5 of this thesis. Fei-Fei *et al.* (2007), for example, incorporate information about previously-learned objects into the prior distribution in an incremental Bayesian model for learning new classes from few examples and Bart and Ullman (2005) make use of features from known classes that are 'similar' to the new class to select relevant features for the new class.

The above models learn a separate classifier for each class, whereas in this work, the aim is to have a single classifier for all classes. This presents the problem that, when a new class is introduced, the system has to learn, from scratch, a new representation based on features derived from all the classes. Being able to reuse or modify features already learned for a given set of object classes, for representing new classes, would increase the adaptability of a single classifier system. This was the motivation, in this work, for having a second stage of feature selection after the *Relief* feature ranking process, during which polygons would be selected one-by-one, from the pool of above-average ranked polygons extracted from the initial repertoire of classes, to derive a representation for each additional class. The approach is loosely based on the idea of Bart and Ullman (2005), that it can be effective to use a feature that has been a successful classifier with a familiar class to help select a suitable feature for a new class.

Another factor worthy of consideration is that, as discussed in Chapter 3, Section 3.4.2, feature ranking techniques such as Relief do not eliminate redundant features. A feature may be relevant in its own right, but when taken in conjunction with other features, it may provide little or no useful additional information and so it may be necessary to apply a technique to detect and eliminate the redundant features.

To implement the incremental approach with the MNIST numerals, and to reduce redundancy in the representation an iterative 'greedy' forward selection method was adopted. The process began with just two classes and a single polygon with which to learn to classify them. Initially, the polygon was instantiated in each training image for the two classes to produce a set of features for use in classification.

The idea was that if classification was sufficiently accurate above a certain threshold, a new class would be introduced into the repertoire, and the system would attempt to classify all three classes using the same single polygon-based representation. If performance deteriorated by a significant amount, a new polygon would be selected from the pool of 'above-average' polygons and used in conjunction with the first in a second attempt to classify the three classes. Then if performance improved sufficiently, another new class would be introduced and the same polygons used to classify all the classes currently in the repertoire. If performance did not improve enough with the new polygon, another new polygon would be chosen to replace it and another attempt at classification would be made. This process would continue until the classification repertoire contained all ten numeral classes.

Each time a new class was added, the existing polygons were instantiated in its associated training examples and stored, and whenever a new polygon was added, its instantiations in the training images of the existing classes were included in the representation. In the nearest-neighbour classification scheme used in this work, this meant that all the newly-generated data had to be stored, which was manageable for just ten numeral classes, but with large numbers of classes would require the application of a technique such as that of Fei-Fei *et al.* or Bart and Ullman, to enable the system to learn new classes from few examples.

A similar incremental 'greedy' forward search approach was employed after the *Relief* feature-ranking stage, with the heterogeneous polygons, applied to the pedestrian data to determine whether it might be possible to build a useful representation for discriminating greyscale images using this type of autonomously-generated feature.

An essential aspect of feature selection is being able to compare features or objects to determine how similar they are, so that, for example, 'like' features can be clustered together, or the classification reliability of features can be tested. Another example is that human vision has to make comparisons between objects, based on their similarities and differences on important attributes, when attention is being directed to relevant objects among irrelevant ones. Section 4.4 discusses the problem of measuring similarity.

4.4 Measuring similarity

Being able to measure reliably how 'similar' two entities are is fundamental to the ability to determine whether a 'new' object instance is the 'same' as a stored representation or concept of an object category or class. Often some sort of distance metric is used and a threshold is set for establishing when objects are sufficiently 'close'.

As discussed in Chapter 2, Section 2.2.2, the process of comparison in biological vision is often modelled in artificial neural networks as the calculation of the dot product of the outputs from neurons in one layer with the stored weights on the afferent connections to a neuron in the next layer and if the weighted sum exceeds the threshold, the receiving neuron 'fires'.

In artificial systems, data clustering techniques for unsupervised object classification or for feature selection also often use Euclidean-based distance measures, such as normalized cross-correlation, which measures the amount of shift between two signals along the various dimensions in a Euclidean representation space.

However, Euclidean distance assumes that the distribution of the data in the pattern space is the same across all dimensions, which may not be the case when the variables or features being

represented by those dimensions are of different kinds, such as, for example, size, orientation and colour. This is the essence of the 'chalk and cheese' problem in measuring similarity (Johnson and Picton, 1995, p39), where in the process of calculating an overall distance, there is often a trade-off between different dimensions. For example, the Euclidean distance between the points A = (1, 8) and B = (4, 4) is the square root of $(1-4)^2 + (8-4)^2 = 5$, but then the distance between A = (1, 8) and C = (6, 8) is also 5, which implies that, for example, differences of 3 units in length and 4 units in orientation between objects A and B can be traded-off against a difference of 5 units in length between objects A and C.

Measurements, such as the Mahalanobis distance, that takes the differences in the spread of data points along different dimensions into account by including, in the Euclidean distance measure, the co-variance matrix of the variables in the calculation, can help to reduce the number of 'chalk-and-cheese' trade-offs.

Also, in work done by Shepard (1964) on human visual attention and perception of similarity, the 'city-block' distance, which sums the absolute differences between measurements on each dimension has been found to be more appropriate than Euclidean distance when object descriptions are based on different types of features, such as size and orientation, rather than, say, on different aspects of an attribute such as colour for which the components hue, saturation, brightness, are harder to separate out. in In the 'chalk and cheese' example above, comparing objects A, B and C, the city-block distance finds the distance between points A and B to be $|1-4| + |8-4| = 7$ and the distance between points A and C to be $|1-6| + |8-8| = 5$, suggesting that A might, in fact, be more like C than B. This illustrates that the similarity between objects measured as a 'distance' depends very much on the features or attributes being considered and the metric chosen.

Another approach is to measure similarity on the basis of presence or absence of salient features. Tversky (1977) proposes that perceived similarity results from feature-matching that weights the shared and distinct features of two stimuli differently and that the overall similarity is evaluated taking into account the saliency of the common features of two stimuli and the

features that are unique to each stimulus. This avoids the problem of different measurement scales along different dimensions.

The similarity measures just described are all deterministic rather than probabilistic, in that when any comparison between entities is repeated, the result is always the same. The methods in this thesis are deterministic and two different approaches are used:

1. Distance metric
2. Feature matching

In the work involving the 'window' features encoded in terms of homogeneous 'light' and 'dark' polygons, and therefore having variable length feature vectors, the 'city-block' distance was used to compare vectors of the same length, comprised of values representative of 'chalk and cheese' dimensions such as greyscale variance of the polygons and their direction from the window centre. The difference measure for the Relief algorithm given in equation (4.1) above, and repeated here in equation (4.2), is the city-block distance between two window instances I_1 and I_2 on the attribute A, normalized by the range of the values for that feature dimension. This measure is also used for finding the nearest neighbours for *Relief* and for classifying the window features. The details are given in Chapter 5, Section 5.6.3. Although the 'city-block' approach has the disadvantage that it relates similarity to distance as in the Euclidean measure, by considering each dimension separately, it avoids some of the problems of trading off 'chalk' against 'cheese' as illustrated the example above.

$$\text{diff}(A, I_1, I_2) = \frac{|\, value(A, I_1) - value(A, I_2) \,|}{max(A) - min(A)} \qquad (4.2)$$

The 'city-block' distance was also used with the heterogeneous polygons and the MNIST numerals data and the NiSIS pedestrian data. Since the polygons were encoded using the sixteen 2x2s patterns shown in Figure 4.5, there is no 'chalk and cheese' issue, so the Euclidean distance could have been used. However, since the Relief algorithm was again being employed for feature selection, the 'normalized city-block' measure seemed an appropriate choice.

For comparing the pixel-pair pattern vectors (Section 4.2.1) a feature matching approach was used to determine their similarity. The corresponding elements in the two vectors were compared, and if they marched, the difference between them was taken to be '0' and if not, the difference was '1'. The overall difference was the sum of all the differences.

The types of features described in this chapter so far have resulted from attempts to make the feature extraction process more autonomous. However, in work that was concerned more with building a multilevel representation rather than autonomous feature extraction, the features were contour 'fragments' obtained from simple hand-drawn 'contour'-based shapes. An example of a shape and its fragments is shown in Figure 4.8 below.

Figure 4.8: A square and its contour fragments used in building a multi-level representation

This work is detailed in Section 4.3.2 and in Chapter 5. What is of interest here, is that the similarity measure for comparing the fragments is an 'all-or-nothing' feature matching approach. Each contour fragment is encoded by a set of different types of features, such as curvature, length, and direction to neighbouring fragments. If two fragments match exactly on all feature values, the difference between them is '0', otherwise the difference is one, so that a fragment is either considered to be present or not with complete certainty. There is no explicit feature-selection, however, in the classification process, the fragments that tend to occur most often in the training objects are examined for a match with input shape fragments before the less commonly occurring ones.

As has been discussed in this section, the similarity between objects is dependent not just on the type of features used to describe them, but also on the multi-dimensional representation space in

which the objects 'exist'. The next section introduces a powerful representation for the 'all-or-nothing' similarity measure described above, based on *hypernetworks* theory (Johnson, 2006).

4.4.1 A hypernetwork framework for representing similarity

Graphs and networks can provide a very useful representation for many types of problem, interactions between people, transport systems and systems for distributing electricity, for example. However, they are limited in that they can only represent binary relationships between pairs of entities, given that an edge, or arc, can only connect two vertices.

In networks, the edge representing the relations between a pair of entities, a and b can be notated as $<a, b>$, connecting the vertices $<a>$ and $<b>$. A relation on these entities can be expressed explicitly in the form $<a, b, R>$, which allows for different relations to be represented, for example $<a, b, R>$ as distinct from $<a, b; R'>$ and also combinations of relations, such as their conjunction or disjunction, to be expressed, for example,

$<a, b; R \wedge R'>$ and $<a, b; R \vee R'>$ respectively.

A *hypernetwork* is the generalization of a network to being able to represent relations among multiple things, through the concept of a *hyper-edge*, often referred to as a *simplex*. Thus, in a *hypernetwork*, the relations among n things, $x_1, x_2, \ldots, x_n$, can be represented by the simplex $<x_1, x_2, \ldots, x_n>$ (Johnson, 2007). To provide visual illustration, a simplex representing n related things can be depicted as a polyhedron with n vertices. A simplex with $n + 1$ vertices is referred to as an n-simplex, for example a 0-simplex is a single point, a 1-simplex is a line and a 2-simplex is a triangle. Figure 4.9 below provides some 'everyday' examples.

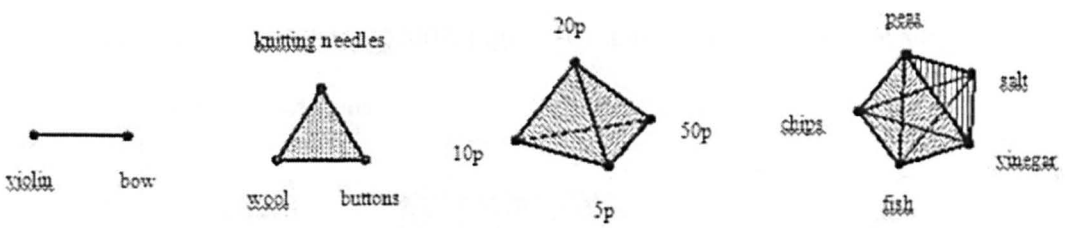


Figure 4.9: Polyhedra showing relations among *n* things
adapted from Johnson, 2007, Figure 1.

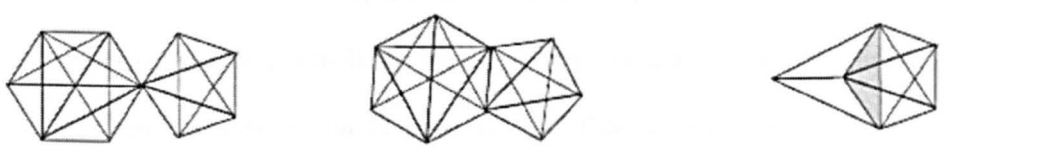
The relation or relations among the elements in the simplex representation can be expressed within the simplex, for example, the 4-simplex in Figure 4.9 above, can be notated as <fish, chips, peas, salt, vinegar; $R_{\text{fish-supper}}$>.

This concept of representing a whole object, W, in terms of its constituent parts, P, that are assembled under a particular relation, R, is the very simple idea that underpins hypernetworks theory (Johnson, 2006). If the relation, R, holds, then the parts will form the object, for example if the fish, chips, peas, salt and vinegar are brought together in one package, then the relation $R_{\text{fish-supper}}$ holds and so the 'fish-supper' object is formed. This process can be written as R: $P \rightarrow W$ and if there are n parts, then R is described as an n-ary relation. This concept, of the formation of objects at one level by assembling parts at a previous level through the application of some kind of relation, is developed further in Section 4.3, in the context of *multilevel* representation.

In this section, the emphasis is on the *multidimensional* relations represented by the vertices and edges of interconnected sets of simplices. A set of simplices is called a *simplicial family* (Johnson, 2006) and a *hypernetwork* is defined to be a *simplicial family* with all its intersecting *faces* (Johnson, 2007).

Let $\sigma_n = <v_0, v_1, \ldots, v_n>$ be a simplex, then $\{v_0, v_1, \ldots, v_m\}$ is its vertex set and the simplex $\sigma_m = <v_0, v_1, \ldots, v_m>$ is a *face* of σ_n iff $\{v_0, v_1, \ldots, v_m\}$ is a subset of $\{v_0, v_1, \ldots, v_n\}$.

If σ_q, a q-dimensional simplex, is a face of both the simplices σ and σ', then σ and σ' are described as being q-near and σ_q is a q-dimensional *shared face* or *q-face* of σ and σ', (Johnson, 2006). Figure 4.10 (from Johnson, 2007, Figure 10) shows some examples of simplices that are connected at different dimensions.



(a) 1 shared vertex (0-near) (b) 2 shared vertices (1-near) (c) 3 shared vertices (2-near)

Figure 4.10: Simplices connected at different dimensions
from Johnson, 2006, Figure 10.

In Figure 4.10(a), the two simplices share a vertex and so they are 0-near. In Figure 4.10(b), the simplices share an edge and are 1-near, while in Figure 4.10(c), the simplices have a triangular shared face and are therefore 2-near.

When multiple shapes share structure, their simplicial representation forms a *star-hub* configuration (Johnson, 2006). This common structure is the intersection of the sets of features comprising the related objects, and can be represented as a *hub*, with the associated objects forming a surrounding *star* of simplices. Figure 4.11 shows a 1-dimensional hub and its star, depicting a pair of features, <e> and <f>, shared by the four objects, <a, e, f>, <b, e, f>, <c, e, f> and <d, e, f>.

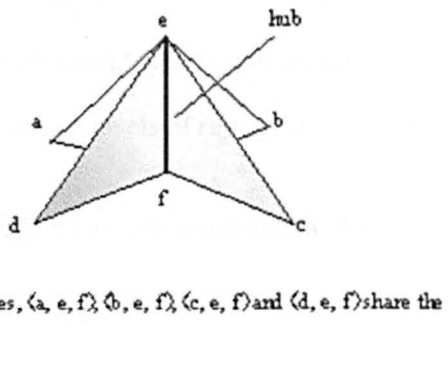


Figure 4.11: A star-hub configuration
after Johnson, 2007, Figure 14

This means that a machine vision system comparing the four objects on the basis of the features <e> and <f> would perceive them as being identical, whereas, the objects could be discriminated if the system were to take the features <a>, <b>, <c> and <d> into consideration.

Therefore the degree of similarity between objects can be measured in terms of the number of shared vertices their simplices have, and this star-hub representation makes it explicit which features are classifying objects as being the same, and which are classifying them as being different, and hence, depending on the requirements of the user, which features are relevant for the task in hand. This is discussed further in Section 4.5.

4.5 Multilevel representation and an adaptable architecture

As discussed in Chapter 2, Section 2, there is considerable evidence to support the hypothesis of Hubel and Wiesel (1962), that the primate visual system is hierarchical, with neurons at successive levels responding to increasingly complex stimuli, but with greater ability to generalize over variations in aspects of appearance such as scale, orientation and position. In Chapter 3, Section 3.7.5, it was found that several machine vision systems have been based on a feed-forward feature hierarchy-based architecture, including that of Serre *et al.* (2005), and LeCun *et al.* (1999), both of which incorporate increasing invariance to image transformations at successive levels, while attaching less importance to the corresponding increase in structural complexity associated with the biological model.

Rather than building invariance to image transformations, the emphasis in this work is on explicit representation of increasingly complex structure at higher levels and on exploring how such structure can 'emerge' to form new levels of representation in a multilevel system.

The thesis explores the problem of forming self-adaptable multilevel architectures through a combination of approaches:

- The principle of grouping or assembling elements at one level under a particular relation to form new constructs at a higher level
- A star-hub approach to intermediate-level classification
- Multilevel feature selection based on 'hub' constructs
- Autonomous modification of the representation in response to the current requirements of the task

These approaches are explored through five sets of experiments, detailed in Chapter 5, that are designed to apply and test various practical techniques for building multilevel representations, in the context of Hypernetworks theory, as described below.

4.5.1 A framework for representing multi-level relations

In Section 4.4.1, Hypernetworks were introduced in the context of their power to represent multidimensional relations by the vertices and edges of interconnected sets of simplices.

In this section, the central idea of hypernetworks theory, that "Wholes are assembled from parts" (Johnson, 2006), is shown to be fundamental to the formation of multiple levels of representation. The concept of a whole object, W, being formed from a set of parts, P, that are assembled under a relation, R, can be written as R: $P \rightarrow W$. If the object has n parts, then R is described as an n-ary relation.

In this way, a new object or structure can be thought of as 'emerging' at a higher level of representation than that of its constituent parts. When vertices exist at one level, the structures that can be formed from them exist at a higher level. So the effect of the relation on the set of parts is to form a simplex at the next level of the hierarchy. Figure 4.12 illustrates this fundamental principle.

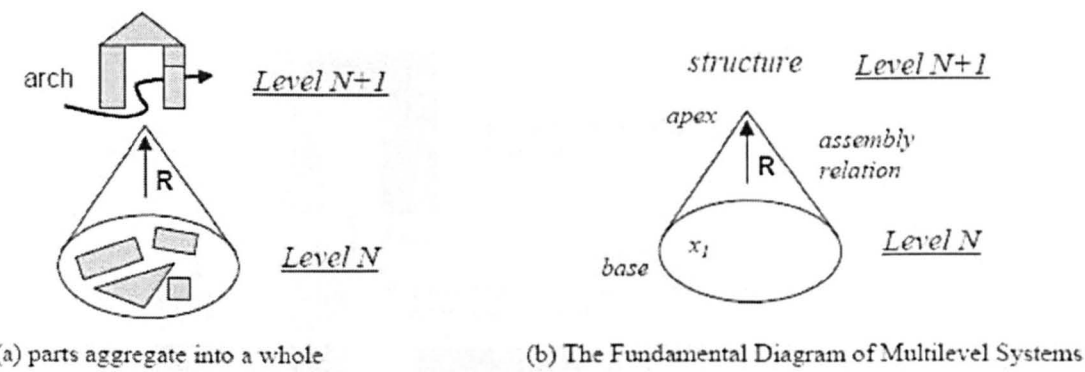


Figure 4.12: The n-ary relation R maps the set of blocks to an arch at the next level
from Johnson, 2006, Figure 16.

In Figure 4.12(a) the set of blocks at *Level N* is assembled under the relation R, to form an arch at *Level* $N + 1$, with the *emergent* property of providing a centre space for things to pass through, as in a door or window, perhaps. Desimone and Duncan (1995) hypothesize that, in human vision, objects in the visual field compete for attention and that attention is an emergent property of the competitive neural mechanisms that work to reduce the ambiguity in the

representation of multiple objects. This is in accordance with the notion of grouping in Gestalt psychology. The collection of parts in Figure 4.12(a) only acquires the 'arch' property when assembled under an appropriate relation. Thus, it is in this way that "the whole is greater than the sum of the parts" (Johnson, 2007). Figure 4.12(b) introduces the "Fundamental Diagram of Multilevel Systems". The initial set is represented by the ellipse at the base of a cone, the assembly relation is applied, and the new structure emerges at the apex of the cone, forming the next level of representation.

The example in Figure 4.13 below illustrates a multilevel hypernetwork representation of the process of forming an arch, assuming the input is from a digital image. Taking individual pixels to be the basic elements at the initial processing stage, *Level* 0, the system might assemble sets of contiguous pixels that are darker than a certain greyscale value, to form, under an *n-ary* relation $R_{\text{RectBlock}}$, 'dark' rectangular blocks of *n* pixels at *Level* 1. It might then configure these dark rectangular blocks, under another *n-ary* relation, in this example, a 3-*ary* relation, R_{arch}, to form an 'arch' at *Level* 2. Thus, when an n-ary relation is applied to the set of elements at *Level* N - 1, the new structure at *Level* N is represented as a simplex.

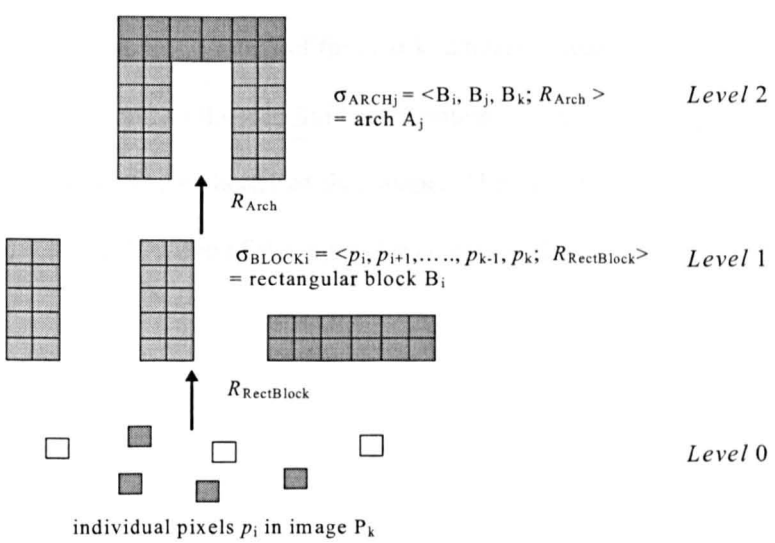


Figure 4.13: A simple *hypernetwork* multilevel architecture
applying *n-ary* relations $R_{(N-1)\text{-}N}$ to assemble sets of elements, $\{e_i, e_{+1}, \ldots, e_k\}$, at *Level* N-1 into more complex structures, $\sigma_i = <e_i, e_{i+1}, \ldots, e_k; R_{(N-1)\text{-}N}>$, at *Level N*.
adapted from Johnson and Sugisaka, 2006, Figures 4 and 7.

4.5.1.1 Lattice hierarchies and multilevel aggregation

This supposes that all the objects in the 'arch' class are the same. However, the 'arch' relation can be applied to different sets of parts and, in addition, different instances of arches can share parts. This is represented by a 'lattice' hierarchy, where the sets at various levels can overlap, Figure 4.14, rather than a tree hierarchy (Johnson, 2006). Thus as structure is assembled from 'parts', two kinds of hierarchical aggregation can be observed:

1) α-, or AND-aggregation, in which the *n-ary* relations require *all* the parts to form the structure at the next level. For example, in Figure 4.14, at *Level* 1, the relation R_1 needs parts b_1, b_2, b_3 and b_4 to form arch A-1 at Level 2. Similarly, the complete sets of parts $\{b_3, b_4, b_5, b_6\}$ and $\{b_6, b_7, b_8\}$ are needed by relations R_2 and R_3 to assemble arches A-2 and A-3, respectively.

2) β-, or OR-aggregation, which partitions sets of structures or objects within a particular type or category into equivalence classes. At *Level* 2 in Figure 4.14, the structures are gathered together to form a set of arches which can be represented as type A-1 *or* type A-2 *or* type A-3.

An example of α-aggregation in this work is assembling sets of horizontally contiguous 'dark' pixels into 'dark runs', as described in Chapter 5, Section 5.3.1.

With regard to β-aggregation, in much of this work, a nearest neighbour approach to classification is adopted and so the n individual members of a class of shape within the training set constitute the n equivalence classes of that shape. Thus an incoming object is classified as being that shape if it matches one of the equivalence classes C_1, *or* C_2, *or*....., *or* C_n.

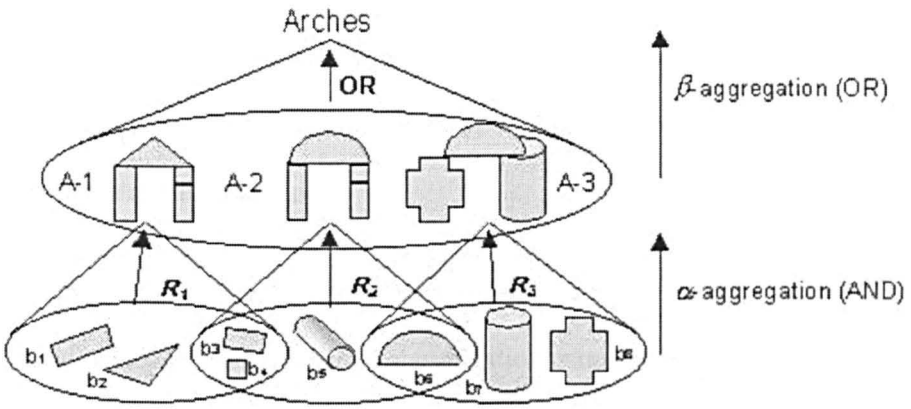


Figure 4.14: Two types of multilevel aggregation from Johnson, 2006, Figure 9

4.5.1.2 The dynamics of networks

The connectivity of networks supports and constrains flows, represented by numbers assigned to the vertices and edges, for example, the flows on the links of a road network might represent numbers of vehicles (Johnson, 2006). Similarly, the connectivity of object parts supports and constrains the flow of information about their spatial relationships. Only connected parts have links between them on which information about their connectivity can be represented, Chapter 5, Section 5.3.

As described in Section 4.5.1, a multilevel architecture can be built through repeated application of the fundamental principle of *hypernetworks* theory, that of parts being assembled under relations to form 'wholes' as shown in the 'Fundamental Diagram of Multilevel Systems', Figure 4.12(b), at each successive level. This is illustrated by the arch example in Figure 4.13.

As well as relations being applied to form new structure at each level, operators act at each level, mapping the associated structure to a number or numbers. In the arch example, the pixels at Level 0 might be mapped to their (x, y)-co-ordinates.

The number or numbers for each entity at the current level can then be 'transmitted' as they are, or through the application of a function to produce a single output, for inclusion in mappings at

subsequent levels (Johnson, 2006) and Chapter 5, Section 5.3.1.2 of the thesis. In the arch example, the pixel co-ordinates might be passed to Level 1 to enable the rectangular blocks, formed under the relation $R_{\text{RectBlock}}$, to be mapped to the values $x_{\min}$, $x_{\max}$, $y_{\min}$, $y_{\max}$, that define them. These values can be thought of as *emergent* properties resulting from the process of forming a rectangular block.

In Chapter 5, Section 5.3, the applicability of the Fundamental Theory of Hypernetworks to visual object representation and classification is investigated, by building, according to its principles, a multilevel representation of simple geometric shapes, namely circles and squares. As well as working with the 'self-similar' aspect of assembling parts to form wholes at multiple levels, the process of mapping structure at each level to numbers that describe its emergent characteristics is explored.

4.5.2 Classifying at the whole-object level

Whole-object classification is the approach taken in the first set of experiments, involving pixel-pairs. The initial pre-processing phase segments the input binary image, locating the 'dark' objects of interest, by applying a spatial relation to the *dark pixels*, at the lowest representation level, to form ordered sets of contiguous dark pixels called *dark runs* at the next level, and then assembles vertically contiguous dark runs to form *dark objects* at the top level. This segmentation process, also employed in the second set of experiments in which the features are contour fragments, is described in detail in Chapter 5, Sections 5.3.1. The objects are then stored as sets of connected vertices, termed 'components' in a graph representation.

At the next stage, the system attempts to extract its own representation of the objects, in this case, a *global* description, with sets of randomly-generated pixel pairs and then classification is on the basis of matching the vectors of pixel-pair patterns representing whole test objects with the vectors representing whole training objects in the database, Section 5.2.3. The low-level features extracted in the later experiments are *local*.

In the multilevel architecture built in Section 5.3, the whole objects, at the top level, are represented in terms of an ordered set of contour fragments of constant curvature, assembled under specific connectivity requirements, Section 5.3.1.3. An object is depicted by a simplex, each vertex of which represents a contour fragment, and classification is based on the degree of overlap of training and test simplices, Figure 4.15.

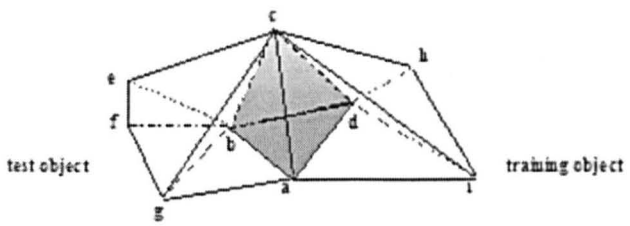


Figure 4.15: Polyhedral representation of the shape simplices of a training and a test object The shaded tetrahedron indicates their shared constructs 'a', 'b', 'c' and 'd'

The test object is assigned to the class of the training example with which it shares the highest number of constructs.

4.5.3 Classifying objects using star-hub analysis of intermediate-level constructs

It has been argued by Lee and Mumford (2003) and in Hochstein and Ahissar's 'Reverse Hierarchy' theory (2002) that, in the primate visual system, lower processing levels continue to provide detailed information to higher levels even after high-level computations have begun, with the high levels 'directing' the lower level processing through feed-back of contextual information, Chapter 2, Section 2.8. Thus, when classification at the whole-object level is inconclusive, a lower level of representation may provide more specific, local information, that can be combined to produce an overall classification decision.

This idea is not new in machine vision, Chapter 3, Section 3.8.7. What is explored in this work is a *hypernetwork* approach to this process. A hypernetworks representation enables the processes of *aggregation*, as illustrated in Figure 4.14, and *disaggregation*, in which scenes or objects are segmented into parts, to be made explicit. Neural networks can provide a powerful multilevel representation, but the weights in the intermediate layers have to be learned from the

training data whereas, in a hypernetworks representation, potentially useful higher-level structure emerges through the application of the fundamental principle.

Instead of focussing on the simplex-representation of objects in terms of their constituent constructs, a system can switch its attention to the conjugate simplex-representation in which each construct is expressed in terms of the training objects in which it appears. These alternative view-points can be represented effectively using an Incidence Matrix, Table 4.2.

Object / Feature	F1	F2	F3	F4	F5	F6
A1	1		1		1	
A2	1			1	1	
A3	1				1	1
A4	1			1	1	
A5		1		1	1	
B1		1		1		1
B2		1	1			1
B3		1		1		1
B4		1		1	1	
B5	1			1		1

Table 4.2: Incidence Matrix for two object classes, A and B, represented by six features, F1 – F6

Each row of the matrix represents an object in terms of the six features, a '1' indicating the co-occurrence of an object and feature. For example, object A1 has features F1, F3 and F5. Each column represents a feature in terms of the objects that share it, for example, F1 is common to objects A1, A2, A3, A4 and B5. Thus there is the object simplex $\sigma(A1) \leftrightarrow <F1, F3, F5>$ and the feature simplex $\sigma(F1) \leftrightarrow <A1, A2, A3, A4, B5>$.

A construct that is shared by more than one object is considered as a 'hub', Section 4.4.1. Hub constructs that occur predominantly in a particular class of object make good classifiers for that class. The feature F5 appears in four out of the five class A objects and in just one of the class B objects, and thus is useful for recognizing objects of type A. Similarly, feature F6 is a potentially useful recognizer for class B. On the other hand, feature F4 is almost equally shared between the two classes, making it less reliable. However, the features F4 and F6 taken together only occur in class B, illustrating the point in Guyon (2008), that a weak feature can be made stronger when in combination with a strong one.

This combination of features can be considered to be at a higher level of representation than that of the single features and other such higher-level structure can be seen in Table 4.2, for example, F4 and F5 that co-occur in objects A2, A4, A5 and B4, and F2, F4 and F6 that appear together in objects B1 and B3. This structure is made explicit in the table through the use of *maximal rectangles* (Johnson, 2006). A *maximal rectangle* encloses the largest rectangular block of 1s for a particular subset of features. The solid outlines indicate the rectangle associated with occurrences of the feature simplex <F2, F4, F6> and the dashed lines indicate the rectangle for the simplex <F4, F5>.

In Chapter 5, Section 5.3.2, an Incidence Matrix is used to show, for the circles and squares data of the second set of experiments, emergent connected 'hub' structure that objects share, both within-class and between-class, at multiple representation levels, Table 5.1, p198. The training objects and the constructs are arranged in the table so that the maximal rectangles make potential higher level structure readily apparent.

The thesis explores three approaches to using intermediate level constructs in classifying test circles and squares.

The first approach is to take into account the dominant class in the object simplex of each individual matched training construct in determining the class of the whole test object, Section 5.3.3.2.1. The second and third approaches involve the use of a heuristic as explained in the next section.

4.5.3.1 Resolving classification conflict using a heuristic

Knowledge about the classification of neighbouring constructs can be used as a heuristic for classifying the current construct under consideration when that construct is shared between classes.

In the second approach to exploring the use of intermediate level constructs, if a test construct is matched with a training construct, the simplex of which contains both circle and square objects, this conflict can be resolved by taking into account the single-class designation of the

construct's immediate neighbours, Section 5.3.3.2.2. This relates to the idea that a weak construct in combination with a stronger one can potentially provide a more reliable classification, as discussed above. For example, in the Incidence Matrix, Table 5.1 (p198), it can be seen that the 10th hub is a mixed-category hub, and that the 11th hub has an entirely 'square' designation, which could be taken jointly with the 10th hub to form an entirely 'square' hub construct at the next representational level.

The third approach requires, in addition to the constraints of the second matching scheme, that the mixed category construct and its single category neighbour appear together in at least one object in the training set, Section 5.3.3.2.3. For instance, the mixed category 3rd hub in Table 5.1 just qualifies, because it appears with the neighbouring all-'circle' 4th hub in circle object 47.

However, as explained in Section 5.3.3.2.2, in order to prevent further conflict, this conversion process is only permitted if the mixed-category construct's neighbour on the opposite side is either also of mixed-category, or is of the same single class of the prospective conversion.

The higher-level constructs resulting from the second approach to conflict resolution may not necessarily be 'emergent' structure within the existing training set, as required for the third approach, in which case, these new constructs could be added to the training set.

The stricter matching requirements of the third approach make use of the 'emergent property' of the existing higher level structure within the training set. In both the second and third approaches, if both the mixed-category construct's neighbours are of the appropriate single-category, an even higher level construct emerges, comprised of three elements. In Figure 5.31 (p209) the mixed-category construct labelled (1, 0, 3, 1) has an all-square construct on either side, labelled (0, 0, 5, 2) and (0, 0, 1, 7) , so that an all-square three-element construct can be formed.

4.5.4 Generalizing the concept of a star-hub representation for inexact construct matching

In Section 4.2.2, a dense sampling approach to feature extraction was described. Neighbouring constructs, in this case, the rectangular window features used in the third set of experiments in a pedestrian recognition task, Chapter 5, Section 5.4, are required to overlap with a shift of just one pixel horizontally or vertically, as shown in Figure 5.34, p215. Feature selection is applied to eliminate irrelevant features and reduce dimensionality, as explained in Section 4.3.3 and then an initial attempt at classification is made at the level of the individual windows. Due to the relatively high variability of the pedestrian and non-pedestrian images, window constructs are less likely to be duplicated in terms of their polygonal representation than the curvature constructs descriptions used in the representation of the circles and squares of the second set of experiments. A variability factor mentioned in Section 4.2.2 is the length of the vector description for a given window across the training images. Hence each window can be thought of as a set of hubs, each of which has, in its star, all the training images of both classes for which it has the same length of description, its instantiations in which can be thought of as having some degree of 'similarity'.

The variability of those same-length descriptions requires a different construct matching scheme. A test window and a training window are considered to be matched when the difference between them, based on the 'normalized' city-block distance, Section 4.4, is the smallest for all the comparisons made within that training window's star. An overall classification of a test image as 'pedestrian' or 'non-pedestrian' is dependent on the class for which the sum of the minimum differences across all the windows is the smaller.

This classification approach was employed both with a training set comprised of pedestrian images alone and with a mixed-category training set, Section 5.4.4.

4.5.5 Using spatial information to constrain the dimensionality of higher-level representations

In Section 4.5.3, the constructs under consideration were each connected to a left and right neighbouring construct in accordance with certain conditions, including the assumption of contiguity, as described in Section 5.3.1.3, and the analysis of higher level structure was restricted to hubs comprised of connected constructs, which is in line with the local nature of connectivity in biological vision systems. Spatial information in the primate ventral visual system is thought to be implicit in the connections between neurons in successive layers, with spatial relations becoming less specific at higher representation levels, where the ability to generalize is greater (Rolls and Deco, 2002, p292).

However, in this thesis, the connectivity is made explicit, as in a constellation architecture, but with *lateral* connectivity rather than the full connectivity of the model of Fei-Fei *et al.* (2007). This looser connectivity provides less information about the relationships of the object parts than the fully-connected model, but has the advantage of reducing computational complexity. The star model of Fergus *et al.* (2005) also has reduced complexity, but object recognition is dependent on the detection of a principle part on which the remaining parts are reliant for their detection. An advantage of the laterally-connected model is that all the parts are of equal importance, which is advantageous for detection purposes, but also for the formation of higher-level structure. The three models are illustrated in Figure 4.16.

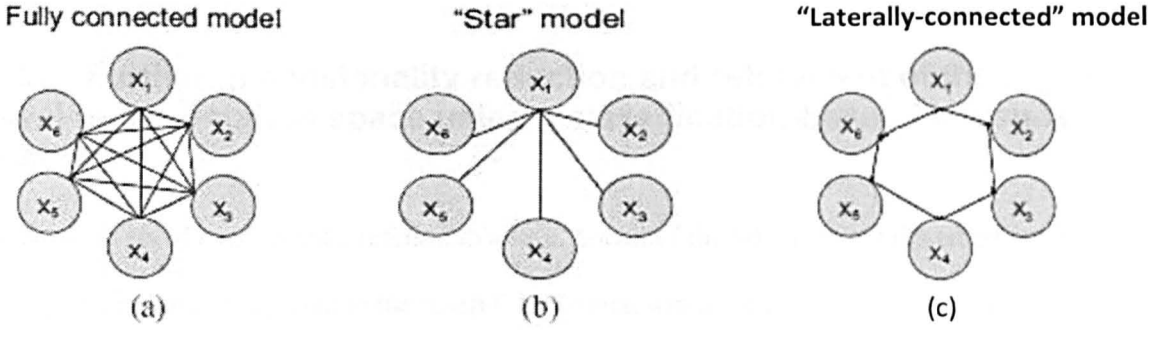


Figure 4.16: Three types of constellation model connectivity
adapted from Fergus *et al.*, 2005, Figure 1: Each node in the three models represents an object part or construct. The simpler connectivity of the star and laterally-connected models in (b) and (c) reduces computational complexity considerably from that of the fully-connected model in (a). Also model (c) is not dependent on a principle part as in model (b).

In the third set of experiments (the pedestrian recognition experiments referred to in the previous section), the connectivity assumption for higher-level structure is that the windows concerned overlap as described above. This limits the formation of higher-level constructs, which, in this work is further restricted to just pairs of windows. Therefore, from among the 76 'useful' windows selected by the *Relief* algorithm (Section 5.4.5.1), only 80 pairs can be formed, instead of the $0.5n(n-1) = 2850$ pairs that would be possible without any spatial constraint.

Each window pair has a variable joint description length, and as with a single window, a pair can be thought of as a hub with all the training images for which it has the same description length in its star. These '2-neighbour' pairs are then used to resolve classification conflict occurring at the single-window level as described in Section 5.4.5.1.

In the fourth and fifth sets of experiments, in which heterogeneous polygons are used to build representations for the MNIST hand-written numerals recognition, and pedestrian recognition, respectively, the spatial constraint applied for forming higher-level constructs is based on proximity. A polygon's location is defined by its centre of mass, and on this basis, a polygon is paired with its closest neighbour. This principle is carried on to successive levels by determining an 'average' of the individual locations for the members of the compound construct at the current level and pairing with the closest neighbour to form the structure for the next level.

4.5.6 Further dimensionality reduction and refinement of the higher-level representation space using a classification-based Incidence Matrix

In the third set of experiments, referred to above, not all of the 80 window pairs emergent under the spatial constraints applied to the 'useful' individual windows are necessarily reliable classifiers and so making use of information about the classification of images in a validation test set can help to eliminate irrelevant constructs and reduce dimensionality in a novel 'wrapper-based' feature selection approach, Chapter 3, Section 3.4.3. An incidence matrix, adapted so that the body of the table shows the number of the training image that provided the

closest match with a test image for a particular window, is illustrated in Chapter 5, Section 5.4.5.

The overarching principle, explained in Section 5.4.5.2, is that indicators of potentially useful higher-level window pair constructs emerge in the Matrix as vertically-aligned pairs of matching non-zero training image numbers. For example, the pair of windows labelled '15' and '16' in the first two rows of Table 5.3 (p223) is a permitted window-pair construct under the spatial constraints defined in the previous section, and it also classifies the test objects labelled '6' and '9' across the top of the Table, through a joint instantiation in training image '4'. This classification is correct, since training object '4' and test objects '6' and '9' are pedestrian images. If such a window-pair is found to jointly classify correctly more often than it misclassifies, it qualifies as potentially 'useful' higher-level structure, the training image instantiations of which can be used in resolving classification conflict occurring at the single window level, as detailed in Section 5.4.5.2.

4.5.7 Adapting the multi-level representation in response to changing user or task requirements

Autonomous machine vision systems need to be able to adapt their behaviour to meet the requirements of the current situation. A system may, for example, look to user input to prompt it to modify its representation architecture, or employ a different classification strategy to improve its performance.

In much of this work, a classification threshold, set by the user, or a reduction in classification performance as task demands increase, triggers the system to either adapt the representation at the current level, by adding another feature, or to attempt classification at a different level, when the classification score is low, Sections 4.3.4, 5.4.5.1, 5.5.3.1 and 5.6.3.

The other trigger employed is awareness of when a classification conflict has occurred, through examination of the star of the matched training hub construct, Sections 4.5.3.1 and 5.5.2.2, or through detection of a disagreement in classification within a higher-level 'compound' hub construct, Sections 4.5.5 and 5.4.5.2.

In addition, a simple classification strategy is adopted, as described in Wolf *et al.* (2006), in which a system can learn the optimal level at which to classify in a specific visual task, through trial and error, Sections 5.5.3.2 and 5.6.4.

4.6 Summary

In this chapter, three key areas of research in machine vision relating to the research questions posed in the thesis were identified, namely, feature extraction, feature selection and the representation architecture. The aim in the thesis is to try, through experimentation in these three areas, to acquire, in the light of progress in the current literature, a better understanding of how artificial object recognition systems might become more autonomous and adaptable.

The chapter described two contrasting approaches to autonomous feature extraction, the first being random, with the requirements for generating what were highly generic constructs being kept as general as possible, and the second being based on algorithmic generation of polygonal features, which gave rise to, in some cases, potentially quite class-specific constructs, that would in themselves be very difficult for a system engineer to 'design'.

With regard to feature selection, conventional approaches to optimizing the representation, including a 'generate-and-test' method and possible ways of limiting the representation to one type of feature were discussed. A feature ranking method using a modified version of the Relief algorithm was described and an adaptation of established approaches to learning new classes from a few examples was explained, the main purpose which was to enable a multi-class classifier to learn a new object class using existing features, thus avoiding having to build a new representation from 'scratch'.

The importance of using an appropriate similarity measure for 'chalk and cheese' systems, for which a Euclidean-type measure would be likely to be unsuitable was discussed. Also, a new approach to measuring the similarity of objects, based on the 'star-hub' analysis of hypernetworks theory to reveal their common structure, was introduced, providing a way of discovering which subsets of features are likely to be reliable classifiers.

The role of the fundamental principle of hypernetworks theory, that 'wholes' are formed from 'parts' under a relation, and in particular that the whole objects 'emerge' at a higher level than that of the parts, was introduced as the basis for the formation of multilevel systems.

The concept of emergence of new structure was presented as key to the ability of systems to self-adapt in response to changes in task requirements. Such structure was described as emerging in different ways – visible in the form of 'maximal rectangles' in an Incidence Matrix, under particular spatial constraints, or implicitly as a result of the conversion of a 'mixed-category' hub construct to the class of its immediate neighbours, or through joint instantiation of a pair of suitably connected constructs in a single training image during classification, as depicted in a classification-based Incidence Matrix.

Finally, simple strategies for enabling systems to adapt their architecture in response to changing task demands were described, including use of the awareness of classification conflict or of failure to exceed a classification threshold to prompt the inclusion of a new feature, or the insertion of a new representation level.

Chapter 5 details the five sets of experiments that explore and test the issues raised in Chapter 4. The first, third, fourth and fifth sets attempt 'autonomous' feature extraction. Feature selection is explored in the third, fourth and fifth sets. Multilevel representation and the concept of an adaptable architecture are investigated in all but the first set.

Chapter 5: Exploring adaptable multilevel representations

5.1 Introduction

This chapter shows the practical application of the approaches discussed in Chapter 4, in the context of five object recognition tasks designed to address the research questions posed at the end of Chapter 3.

Section 5.2 describes the first set of experiments, in which the first of the two approaches to autonomous feature extraction adopted in this work is explored. In Section 5.3, the second set of experiments is divided into three phases. In Phase 1, (Section 5.3.1) the hypernetworks-based multilevel architecture is investigated, in Phase 2, (Section 5.3.2) the emergence of potential higher-level structure is demonstrated using an Incidence Matrix, and in Phase 3, (Section 5.3.3) object recognition at multiple levels of representation is attempted. The data used the first two sets of experiments consists of simple, hand-drawn geometric shapes. Section 5.4 covers the third set of experiments in which an 'overlapping window' approach to feature extraction is adopted and a modified version of a standard feature selection technique is applied in a pedestrian recognition task. The abstraction of potential higher-level structure through analysis of a 'Classification Incidence Matrix' is also detailed. In Sections 5.5 and 5.6, in the fourth and fifth sets of experiments, the second approach to autonomous feature extraction, investigated in the context of hand-written numeral and pedestrian recognition tasks, respectively, is explained. These two sections also describe the incremental formation of a multilevel representation.

5.2 First set of experiments: randomly-selected pixel-pair features

5.2.1 The dataset

The circle, diamond and square shapes used in these experiments are hand-drawn as contours to make the drawing process of generating data easier and more efficient. The system then detects and interprets them as silhouettes, as described below. There are just three single template training shapes, one for each class, as shown in Figure 5.1, and eighty-eight test examples of each of the three classes, Figure 5.2.

Figure 5.1: The template training shapes (not to scale)

Figure 5.2: The 88-strong test sets of circles, diamonds and squares (not to scale)

5.2.2 Detecting and scaling the shapes

As each row of an input image is scanned, pixels of greyscale value < 240, are taken to be 'dark' and hence foreground pixels and those of greyscale between 240 and 255 are taken to be background pixels, (the threshold of 240 having been found, by inspection of the images, to give a good segmentation of the shapes from the background). Along each row, contiguous 'dark' pixels are assembled to form runs and are stored as the vertices of a graph that represents the whole image. Then, the algorithm inserts an edge between pairs of vertices that correspond to vertically contiguous runs that overlap by a margin of at least one pixel. Once all the edges have been added to the graph, a recursive process is applied to finding all the connected components in the graph thus locating all the shapes in the image. The shapes are stored as sets

of 'long runs' that are formed by horizontally connecting the outside end pixels of the shape on each row. Each shape is then encased in a bounding box and is scaled to a 75x75 pixel image.

5.2.3 Encoding the shapes

A shape is represented as a vector of the numbers that form the sequence of pixel pair configuration patterns that occur at the locations specified by a randomly-generated set of pairs of (x, y)-coordinates. As described in Chapter 4, Section 4.2.1, and Figure 4.1 (p129), there are four possible configuration patterns for the pixel pairs. These are labelled '0' when both the selected pixels are background, '1', when the first pixel is foreground and the second is background, '2' when the first pixel is background and the second is foreground, and '3' when both pixels are foreground.

In the first experiment, ten different sets of sixty random pixel-pairs are generated (Appendix A, Table A.1), so for each set, the shapes are encoded as 60-dimensional vectors consisting of 0s, 1s, 2s, and 3s. Figure 5.3 shows two instances of a 'template map', the different colours in which indicate the regions of overlap of the three training template shapes when they are superimposed in one image. The numbers 0 to 3 are used in the Figure to indicate how many of the shapes overlap at a given location. In addition, on each template map, a different set of random points is picked out in black, '*' for the first pixel selected in a pair, '#' for the second. Map(a) displays the second set of random points, and Map(b) the third, from Table A1. To avoid clutter, the actual pairings are not shown.

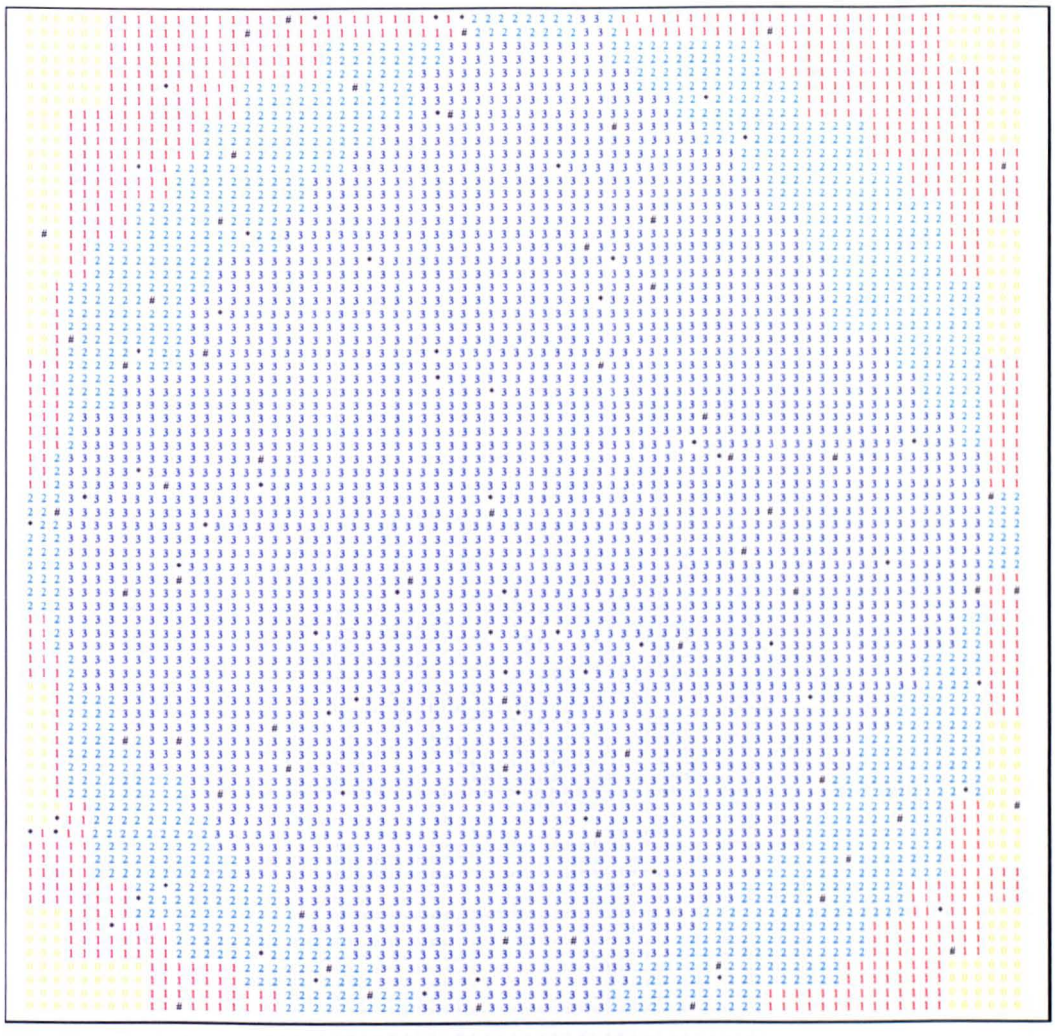


Figure 5.3: Template Map(a)
(formed from the 3 objects in Figure 5.1)

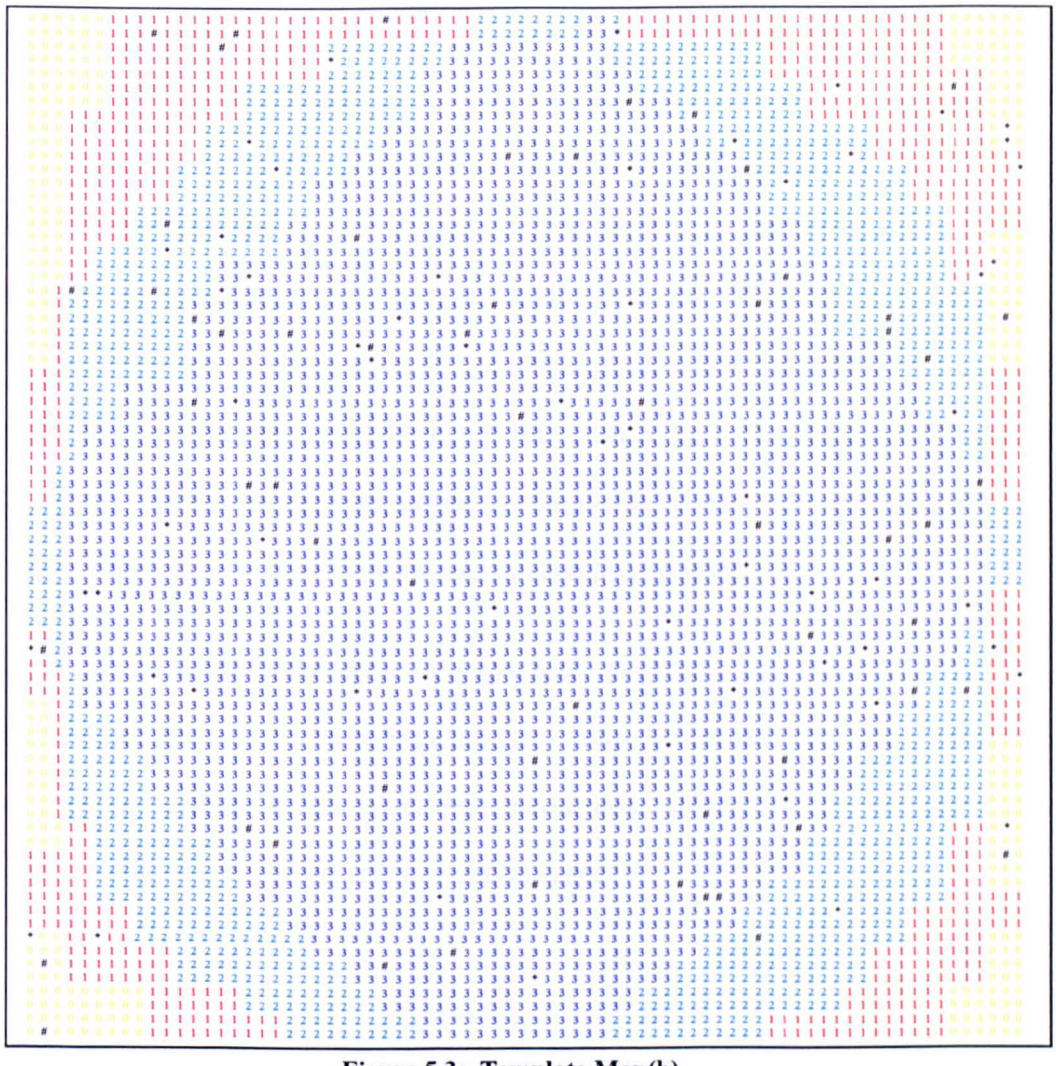


Figure 5.3: Template Map(b)

Figure 5.3: Template maps showing shape overlap and random pixel-pair points
(ends of pixel pairs ar e shown as #)

A classification is made by counting up the number of mismatches between the test sample vectors and those of each of the three templates, the shape being assigned to the class for which the lowest mismatch is found. If two or more shapes have the same score, a non-classification is made. A correct classification adds one point and a misclassification and a non-classification both add zero to the overall recognition score. In Table A2 (Appendix A), scores of less than 100% are highlighted in red and in subsequent tables, if a score of 100% is an improvement over the corresponding result in Table A2, it appears in blue. If a 100% score is the same as the result in Table A2, it appears in black. An improved score that is less than 100% is denoted in green, while a reduced score is shown in purple. Tables A2 – A14 (Appendix A) also indicate the average error across the three shapes and the average numbers of the different configurations selected for each class of shape with each random pixel-pair set.

Figure 5.4 shows the three test-sets as output from the classification system in response to the third dataset. Each boxed shape has a miniature shape in the lower left corner, which, if it matches the big shape, indicates a correct classification. If more than one miniature shape appears in the box, a non-classification has occurred.

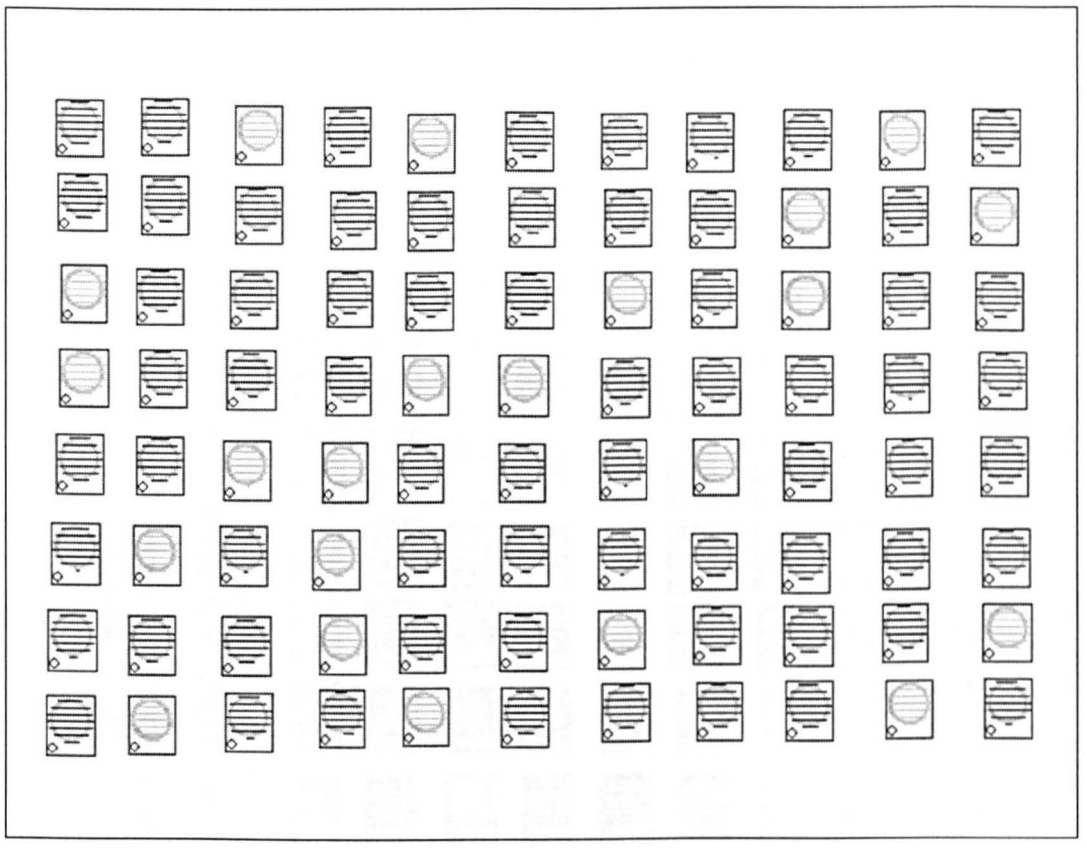


Figure 5.4(a) Circles with random pixel-pair set 3
(the horizontal lines indicate that the shapes are filled in with black pixels)

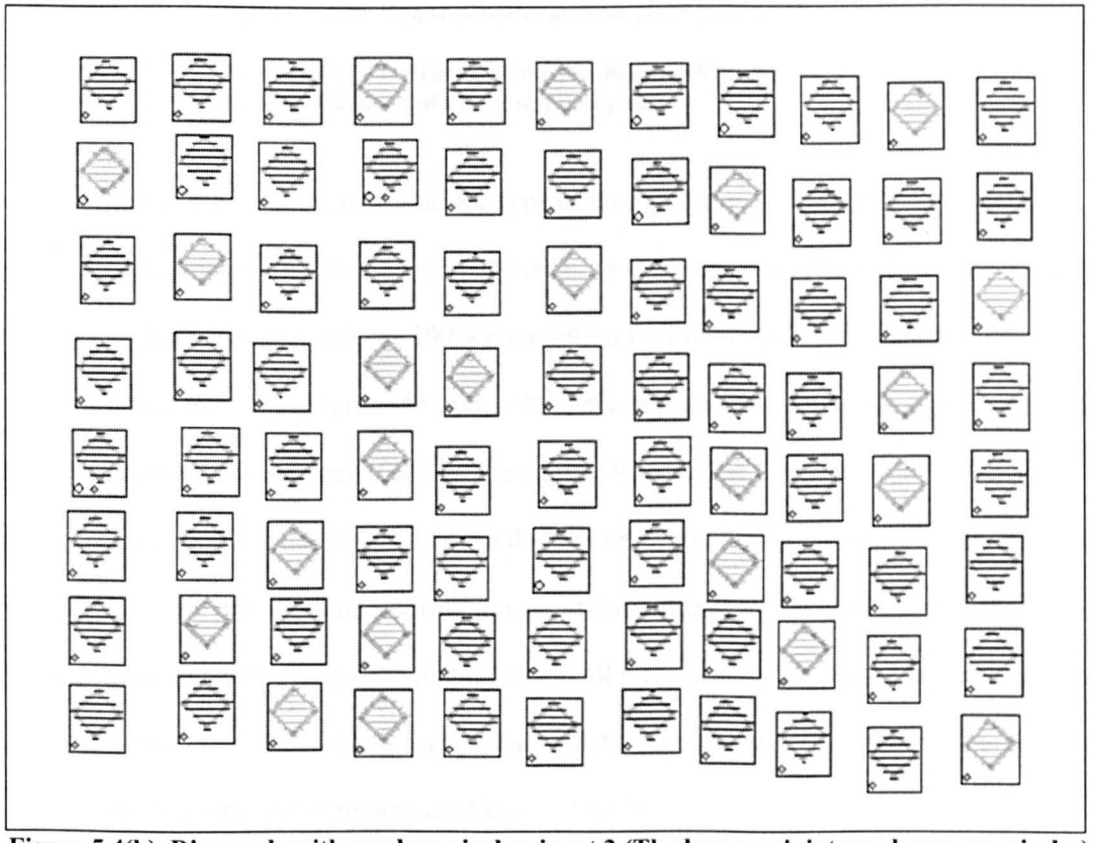


Figure 5.4(b) Diamonds with random pixel-pair set 3 (The larger miniature shapes are circles)

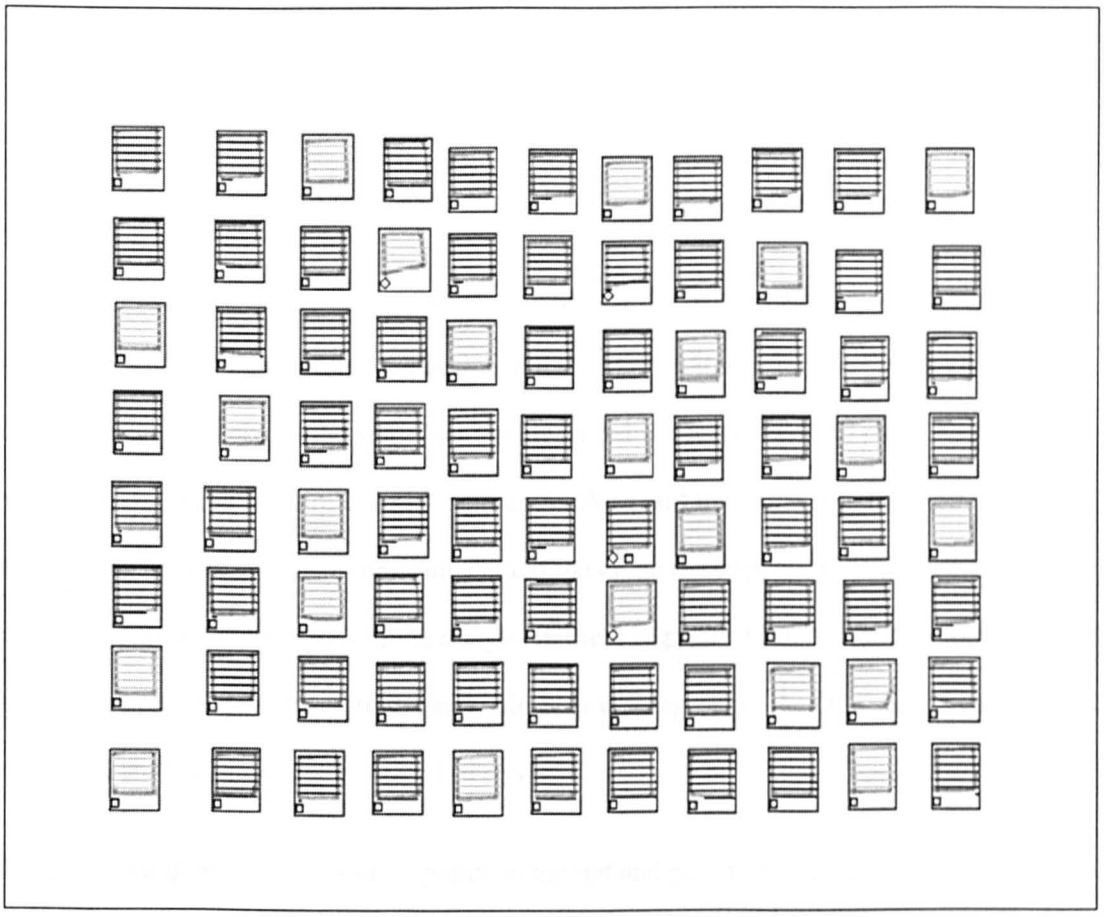


Figure 5.4(c) Squares with random pixel-pair set 3

Figure 5.4: Classification output for the three test sets with the third set of random pixel-pairs

Not surprisingly, the location as well as the type of configuration of the random pairs affects performance, as does the number of pairs selected. In Template Map (a), showing the second set, with which the system achieves 100% recognition for all three shapes, there are fewer points within the yellow background regions than in Template Map (b) which depicts the third set, for which the system scores 100% on the circles, 91% on the diamonds and 95% on the squares. This will affect the results to some degree, since points that are in the background for all three shapes are providing no information about the differences between the shapes. The same applies to the dark-blue points that belong to all three shapes. For these particular shapes, the region where they all overlap is quite large, which means there is a lot of redundancy. Map (a) has about 73 pixels in this region and Map (b) has 75.

Thus, it is only pixels selected from within the red and pale-blue regions (with one and two overlapping shapes respectively) of the template maps that are providing any discriminatory information about the three shapes and indeed any non-redundant information about their within-class similarities.

To illustrate the kind of information being provided by the red and pale-blue regions, consider when one pixel of the pair is in a red region and the other is in a pale-blue region. Most often, if both are foreground pixels, the shape is a square, if red is background and pale-blue is foreground the shape is a circle, and if both are background the shape is a diamond. An exception to this occurs where the diamond and the circle overlap, but not the square – mainly at the middle part of the two sides of the image. In these regions, if both pixels are foreground, the shape is a circle, if red is background and pale-blue is foreground, then the shape is a diamond, and if both are background, the shape is a square.

The second random set has about 39 points in the red and pale-blue regions, whereas the third set has just 31, so it appears that this factor is affecting the results. When the number of random pairs is increased to 100, the performance with the third random set is considerably improved (Table A5, Appendix A).

5.2.4 Restricting the type of configuration selected

The next experiment is concerned with the relative 'usefulness' of the different configurations, firstly configuration '3' alone, with the other three configurations counting as 'not 3', and then configurations '1' and '2' combined, with '0' and '3' counting as 'not 1 and 2'.

The results in Table A3 (Appendix A), for '3' and 'not 3', using 60 random pairs, as before, show improvement in all the scores that were under 100% in the first set of experiments, and no 'worsening' of any full 100%-scores. The table indicates that, as one would expect, considering the shapes being used, the larger proportion of pairs are of type '3' configuration for all three shapes. So these pairs are being generated in sufficient quantity to give reasonably good

recognition. The third random pixel-pair set fares considerably better with the diamonds than when all four configurations are being taken into account – 99% as opposed to 91%.

This improvement is probably due to the fact that having just two configurations instead of four leads to a reduction in the potential for error. The input vectors for the full four configurations are comprised of an assortment of '0s', 1s', '2s' and '3s' and for every mismatch an error is counted, so with only two configurations, '3s' and ' not 3s', some of the scope for error is lost. Thus discriminatory power is reduced but the ability to generalize is correspondingly enhanced.

Increasing the number of random pixel-pairs to 100 (Table A4, Appendix A) further improves performance over the original 60-pair, all-configurations results. Also this result is slightly better than for 100 pixel-pairs with all four configurations (Table A5, Appendix A), which confirms the importance of the increase in the ability to generalize in the present circumstances.

In the restriction of the permitted configurations to types '1' and '2' and 'not 1 and 2', there are fewer of type '1 and 2' than of type '3' within each of the 60-pair random sets and, as a result, performance is correspondingly poorer. Table A6 (Appendix A) shows that, while there are some slight improvements with recognition of diamonds with sets '1'and '3', there is also some deterioration in recognition with sets '3', '4', '5', '7', '9' and '10'. Another factor which is possibly affecting performance is that the 'not 1 and 2' configurations, '0' and '3', are opposites – background/background versus foreground/foreground, so that grouping them together into one type may reduce discriminatory ability considerably. In other words, it seems that information about 'edges' and 'not edges' without further categorization of the 'not edges' as 'shape' or 'background' is insufficient for reliable recognition.

Selecting 100 random pairs increases the quantity of type '1' and '2' configurations and performance improves correspondingly. Table A7 (Appendix A) indicates that there are higher scores than in the original 60 random-pairs, all-four configurations experiments for several of the larger random sets, but that the fourth set does rather less well with the diamonds. However, the results are still not as good as for 60 random-pairs with the '3' and 'not 3' configurations, Table A3.

Increasing to 130 pairs brings further improvement – 100% recognition for all but the diamonds, with which the third random set gives a marked deterioration (Table A8, Appendix A). So, in general, even when the number of points is increased in order to produce more type '1' and '2' pairs, recognition is not as reliable as for all four configurations or types '3' and 'not 3'.

5.2.5 Restricting the distance between paired pixels

It appears that restricting the permitted distance between the pixels in each pair adversely affects performance for a given number of random pairs. Tables A9, A10 and A11 (Appendix A) show that, as the distance, in both horizontal and vertical directions, is reduced from <=10 to <=5 and finally to <=3 (using sets of 60 pixel-pairs) recognition deteriorates correspondingly.

Increasing to 150 pairs, for all configurations and the distance set at <=10 gives 100% recognition for all except the diamonds with the fifth random set (Table A12, Appendix A).

However, with 150 pairs, all configurations and any distance, all shapes are correctly classified across all ten random set (Table A13, Appendix A), and the same result is obtained for a second group of ten sets (Table A14, Appendix A).

One might expect that restricting the distance would provide useful 'local' discriminatory information. Possibly the reason this does not appear to be so is that, with a relatively small selection of pixel-pairs, the amount of redundancy, due to pairs being chosen within regions where all the shapes overlap, is increased to the extent that essential discriminatory information becomes scarce. Of course, when more random pairs are generated, there is a better chance of gathering enough useful information to compensate for this.

5.2.6 Conclusions from the first set of experiments

The main conclusion from these experiments is that the randomly generated configurations are remarkably effective for pattern recognition, and this is very encouraging for this research, since it is a step towards vision systems that can generate their own constructs.

5.3 Second set of experiments: Investigating multilevel representation using hypernetworks

The second set of experiments builds, step-by-step, a multilevel structural representation of sets of simple shapes and then tries to classify them. As in biological vision, the system is based on the assumption that a single visual processing layer is insufficient for solving the problem of invariant object recognition.

The aim is to determine whether the multilevel combinatorial approach of a hypernetwork representation, and its associated notation, can be used to describe visual objects, so that new examples of objects belonging to familiar categories can be correctly classified, and objects not belonging to the known classes will elicit a lower recognition response.

The experiments are divided into three phases:

Phase 1 (Section 5.3.1) explores the multilevel architecture.
Phase 2 (Section 5.3.2) investigates representation of structure with hypernetworks.
Phase 3 (Section 5.3.3) explores object recognition using structure at different levels of representation.

5.3.1 Second set of experiments - Phase 1: The multilevel architecture

This section describes the various processing stages of the multilevel system, from the initial input image to the representation of whole objects, as it maps sets of 'entities' at one level to labelled objects or structures at the next.

5.3.1.1 Image segmentation

There are two preprocessing stages for segmenting the image and locating the objects of interest, Figure 5.5:

1) At level 0, the input is the set of image pixels, $p_i \in P_k$, $(0 <= i < k)$. They are processed with a binarization relation, $Pre_{0,1}$, that assembles horizontally contiguous sets of pixels, of greyscale value $< = 240$, into dark runs, b_i, at *Level* 1, Figure 5.5. The angled brackets in the figure depict the formation of a new entity at *Level* $N + 1$ by assembling a set of

elements under a given relation at *Level N*. The assembled dark runs are then stored as the vertices of an image graph.

2) At level 1, the proximity relation, $Pre_{1\text{-}2}$, adds an edge to the graph between pairs of runs that are sufficiently close. In this work, 'sufficiently close' means that the runs are vertically contiguous and that the right end of the first of the pair of runs is at least eight-connected to the left end of the second, or vice versa. This process forms connected components in the graph, which the system identifies through a recursive search for all the subsets of connected vertices, and these components represent the dark objects that appear at *Level* 2 in Figure 5.5.

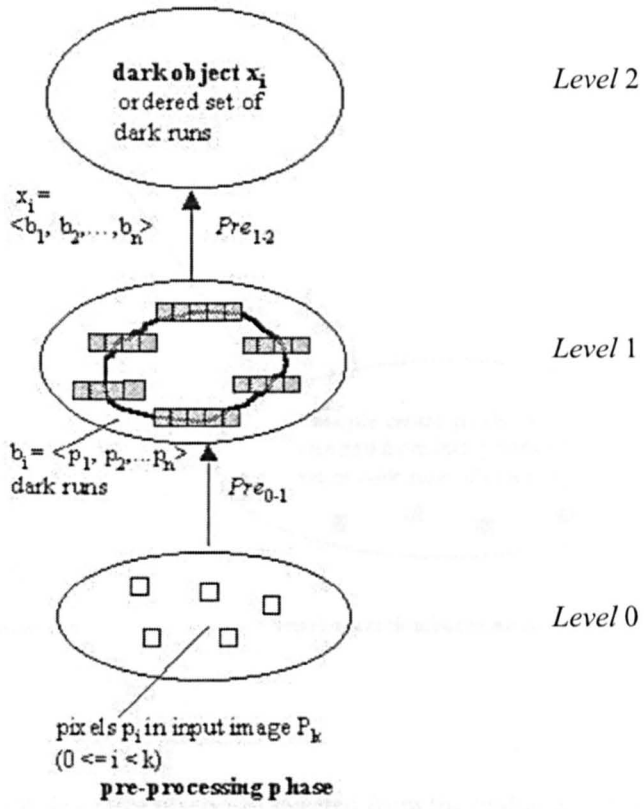


Figure 5.5: Dark pixels form dark runs which are then assembled to form dark objects

5.3.1.2 *Multilevel shape representation*

5.3.1.2.1 Level 0

For each dark object, x_i, an ordered set of dark runs is returned for processing at Level 0 of the representation/recognition phase. At this level, a set of sample centre pixels is extracted from the midpoints of the set of dark runs, Figure 5.6.

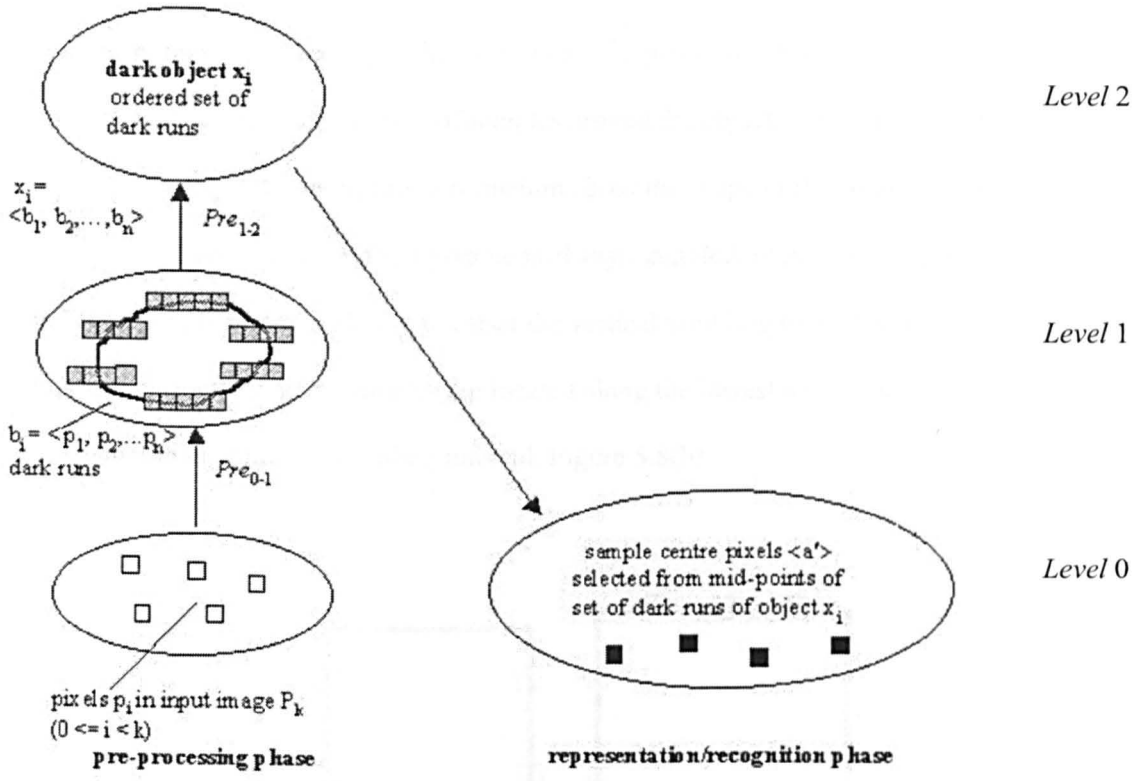


Figure 5.6: A set of sample centre pixels are selected from the midpoints of the set of dark runs from the newly-formed dark object, x_i

The midpoint of the first dark run in the set – the run with the lowest y-value – is always selected as the first sample centre, with the subsequent sample centres being chosen at a preset sampling interval. Figure 5.7 shows a set of $n \times n$-pixel samples, each centred on a selected pixel, a_i.

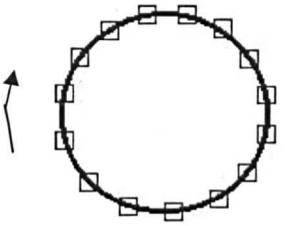


Figure 5.7: Selecting the sample centres round an object

The sampling interval setting in this work is S = 23 pixels, a value found, by experiment, to provide a satisfactory distribution of samples around the object, with samples being taken sufficiently frequently to capture information about the shape of the contour. It is also ensured that the dark run with the highest y-value is always sampled, to prevent the problem illustrated in Figure 5.8(a), and in addition, to offset the vertical sampling bias of only scanning the input images horizontally, extra samples are located along the lowest and highest of the runs if they are longer than twice the sampling interval, Figure 5.8(b).

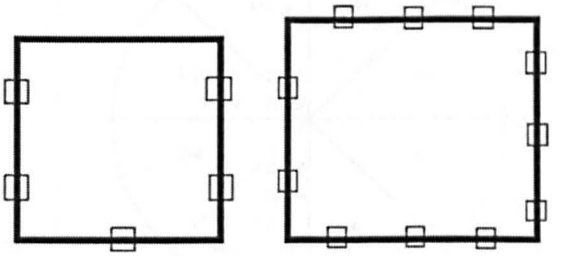


Figure 5.8: Compensating for the vertical sampling bias
(a) Absence of sample across the top of the shape due to the sampling interval
(b) Offsetting the vertical sampling bias by taking extra samples from 'long' dark runs across top and bottom of the shape

Also at Level 0, the sample centre pixels are mapped to the 'generalised' direction to both their left and right next-but-one neighbours. In Figure 5.9, the sample centre pixel, a′, is shown at the base of the shape, with the *generalised directions* d_{left} and d_{right} to the left and right next-but-one sample centres respectively, indicated in grey. In this example, d_{left} maps to the value D3, while d_{right} maps to D0.

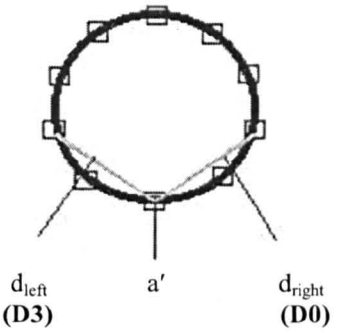


Figure 5.9: Mapping sample centre pixel, a', to the 'generalized' directions of its left and right next-but-one neighbours, 'D3' and 'D0', respectively

These directional values belong to the set {D0, D1,...., D7}, a set of *direction sectors* from the *generalised direction wheel*, Figure 5.10, in which the *direction sector* to which a particular sample point is assigned depends on the gradient of the straight line connecting it to the second point of interest, as measured in terms of the horizontal change in position, *dx*, and the vertical change in position, *dy*, Figure 5.11. These directional mappings are then stored for use at a later stage of processing.

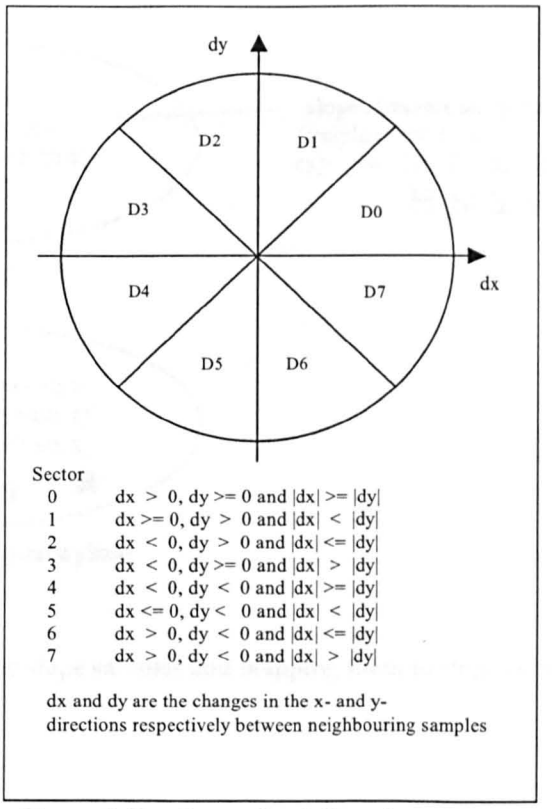


Figure 5.10: Generalized direction wheel
The wheel's centre is taken as the 'origin' of the '*dx*' and '*dy*' axes, with '*dx*' positive to the right of the origin and negative to the left, and '*dy*' positive above the origin and negative below.

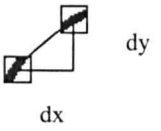


Figure 5.11: Finding the change in x- and y-positions between sample-centres

5.3.1.2.2 Level 1

In order to represent the sampling of the object x_i at the points a'_i selected at *Level* 0, a relation $R_{0\text{-}1}$ is applied to assemble 13x13 blocks of pixels, $bb(a'_i) = \langle a^*_0, a^*_i, \ldots \rangle$, centred on each a'_i, to form a *slope sample*, $\langle s_i \rangle$, at *Level* 1, Figure 5.12. In addition, Figure 5.11 shows the mapping, at *Level* 1, of a *slope sample*, $\langle s_i \rangle$, to a *slope element* in the set {S0, S1,....., S5}, each member of which is a label for a particular *slope template* against which the slope sample is matched. An example of the labelling of slope samples is shown in Figure 5.13.

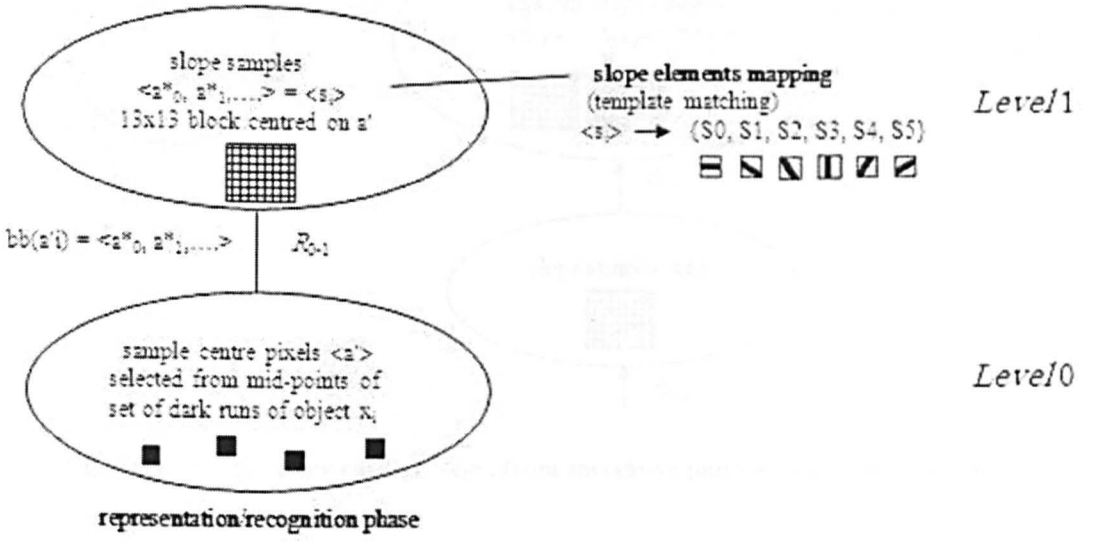


Figure 5.12: Forming slope samples and mapping them to slope elements at Level 1

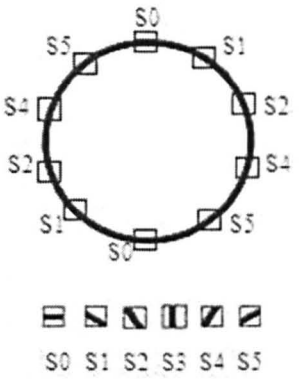


Figure 5.13: Template matching and labelling slope samples

5.3.1.2.3 Level 2

The structure at *Level* 2 is assembled under relation $R_{1\text{-}2}$ to form successive pairs of adjacent slope samples, $s_{i\text{-left}}$ and $s_{j\text{-right}}$, into *curvature entities*, $<c_i>$, Figure 5.14.

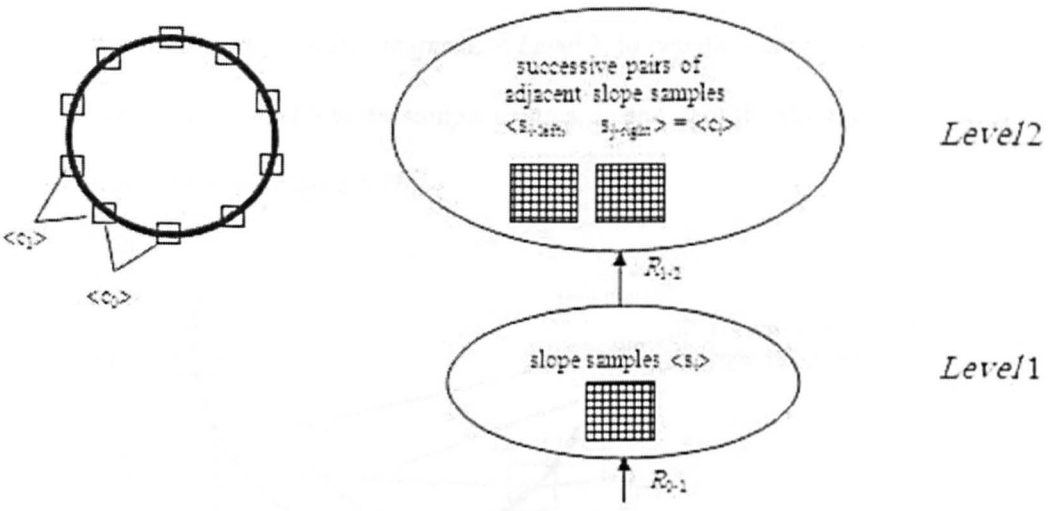


Figure 5.14: Forming curvature entities, $<c_i>$, from successive pairs of adjacent slope samples

Also at *Level* 2, these curvature entities are mapped to *curvature elements* –

$<c_i> \rightarrow$ {C0, C1, C2, C3} - by means of a look-up table, Figure 5.15.

	S0	S1	S2	S3	S4	S5
S0	C0	C1	C2	C3	C2	C1
S1	C1	C0	C1	C2	C3	C2
S2	C2	C1	C0	C1	C2	C3
S3	C3	C2	C1	C0	C1	C2
S4	C2	C3	C2	C1	C0	C1
S5	C1	C2	C3	C2	C1	C0

Figure 5.15: Mapping curvature entities to curvature elements

From the table in Figure 5.15, it can be seen that pairs of slope samples with the same slope map to C0. C1 through C3 indicates increasing slope difference between the samples, with C3 being the value assigned when there is maximum difference.

Each curvature entity, $<c_i>$, is also mapped, at *Level* 2, to two directional values which were previously assigned at *Level* 0 to the sample centres, a'_i and a'_j of the slope samples $<s_{i\text{-left}}>$ and $<s_{j\text{-right}}>$ comprising $<c_i>$, Figure 5.16.

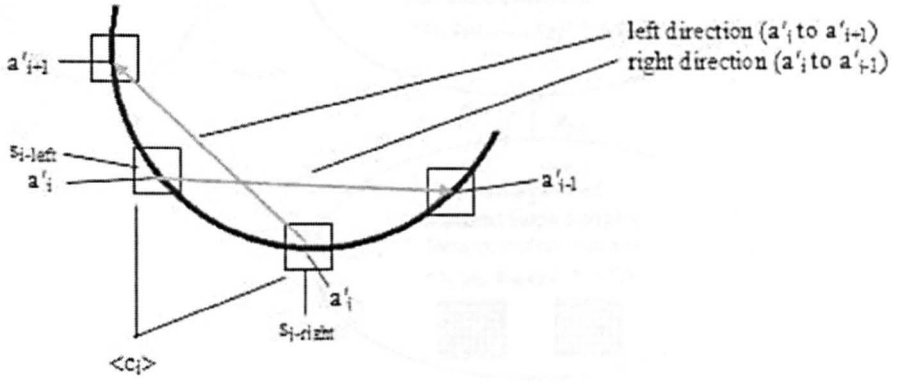


Figure 5.16: Assigning left and right directions to a curvature entity, $<c_i>$, at Level 2

In the figure, $<c_i>$ is mapped to the *right* direction of sample centre a'_i: $<c_i> \longrightarrow d_{right}(a'_i)$, and $<c_i>$ is mapped to the *left* direction of sample centre, a'_j: $<c_i> \longrightarrow d_{left}(a'_j)$. In other words, $<c_i>$ takes the direction from a'_j to a'_{i+1} as its left direction and the direction from a'_i to a'_{j-1} as its

right direction. In this example, the assigned left and right directions would be $D2_{L2}$ and $D7_{L2}$, repectively, from the direction wheel in Figure 5.10. The additional subscript, 'L2', indicates that these directional mappings are occurring at *Level* 2.

5.3.1.2.4 Level 3

At *Level* 3, sets of the curvature entities, or pairs of adjacent slope samples from *Level* 2 are assembled under a curvature relation, $R_{2\text{-}3}$, that groups curvature entities into different types of *curvature construct* according to whether they from straight line segments, line segments of constant positive curvature, or line segments of non-constant curvature. Each *curvature construct*, $<cc_i>$, is then mapped to a curvature label:

$$<cc_i> \longrightarrow \{CC0, CC1, CC2\} \tag{5.1}$$

where CC0 = constant 0-curvature, CC1 = constant positive curvature and CC2 = non-constant curvature, Figure 5.17.

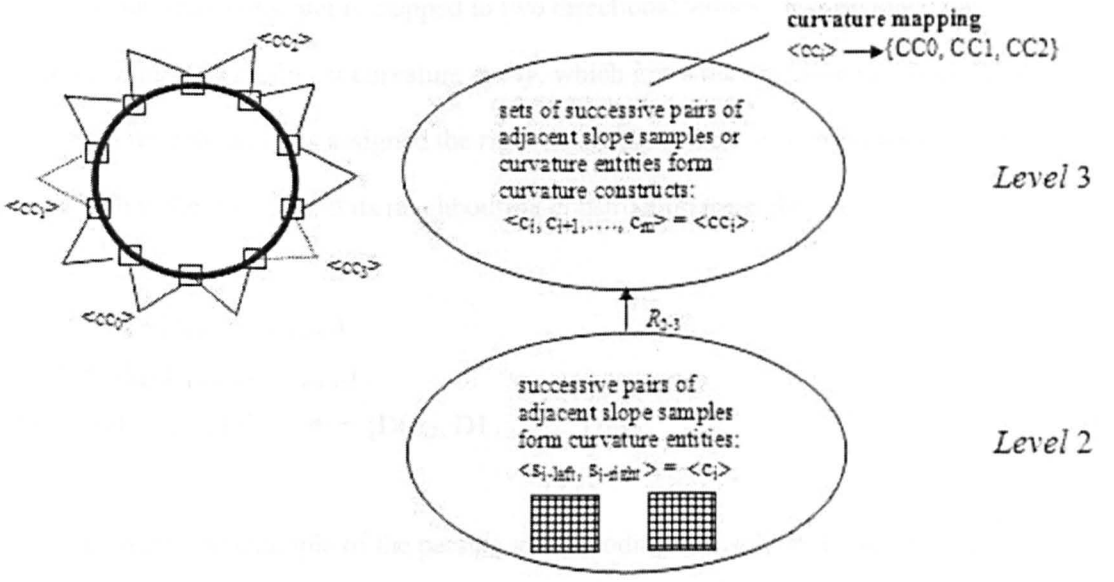


Figure 5.17: Assembling curvature entities at Level 2 to form curvature constructs at Level 3 from Figure 5.21

In addition, at Level 3, the curvature constructs are each mapped to a length-ratio value. The first stage of the length mapping is the application of a formula that expresses the number of

curvature elements in the curvature construct as a proportion of the total number of curvature elements comprising the whole object to which the construct belongs, (5.2).

$$l(cc_i) = \frac{\#\text{ construct curvature elements}}{\#\text{ whole object curvature elements}} \tag{5.2}$$

where $l()$ is the required length ratio and $<cc_i>$ is the curvature construct in question. The resulting ratio is then assigned to one of four length-ratio 'bands', (5.3).

$$\begin{aligned} &L0 = (1 - 25)\% \\ &L1 = (26 - 50)\% \\ &L2 = (51 - 75)\% \\ &L3 = (76 - 100)\% \end{aligned} \tag{5.3}$$

Thus the length mapping can be written as

$$<cc_i> \longrightarrow \; l(cc_i) \in \{L0, L1, L2, L3\}. \tag{5.4}$$

Also, each curvature construct is mapped to two directional values. It is assigned the left directional value of its leftmost curvature entity, which gives the direction to its neighbouring construct on the left, and it is assigned the right directional value of its rightmost curvature entity, which is the direction to its neighbouring construct on the right:

$$\begin{aligned} &<cc_i> \longrightarrow d_{left}\,(c_{\text{left end of construct}}) \\ &<cc_i> \longrightarrow d_{right}(c_{\text{right end of construct}}) \\ &\in \{D0_{L3}, D1_{L3}, \ldots\ldots, D7_{L3}\} \longleftarrow \{D0_{L2}, D1_{L2}, \ldots\ldots, D7_{L2}\} \end{aligned} \tag{5.5}$$

Figure 5.18 shows an example of the parsing and encoding of a square, Object 0 from the training set, Figure 5.25, into its set of constituent curvature constructs.

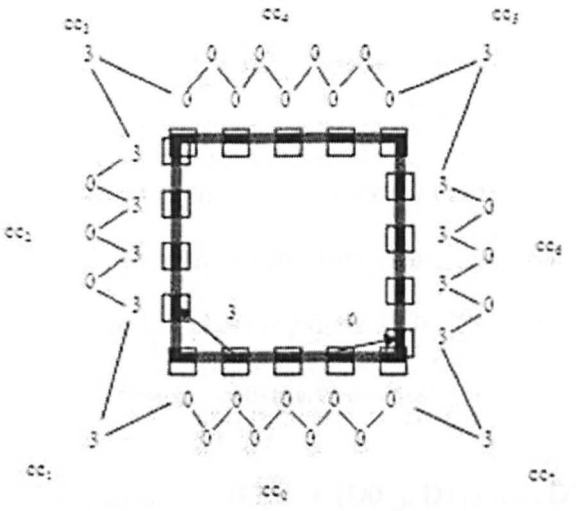


Figure 5.18: Parsing a square into a set of curvature constructs

To reduce clutter in the figure, only the numerical part of the slope and curvature labels has been included. The inner numbers represent the slope elements {S0, S1,….,S5}, to which the eighteen slope samples around the shape have been mapped at *Level* 1, Figure 5.12, while the outer values represent the curvature elements, {C0, C1, C2, C3}, that result from combining successive pairs of adjacent slope elements in the look-up table at *Level* 2, Figures 5.14 and 5.15. The curvature mapping at *Level* 3, $<cc_i> \longrightarrow$ {CC0, CC1, CC2}, Figure 5.18, parses the shape into eight sets, cc_i,of curvature elements. Starting at the bottom of the figure and proceeding clockwise we have:

$cc_0 = \{0, 0, 0, 0\}$; $cc_1 = \{3\}$; $cc_2 = \{0, 0, 0\}$; $cc_3 = \{3\}$; $cc_4 = \{0, 0, 0, 0\}$; $cc_5 = \{3\}$;

$cc_6 = \{0, 0, 0\}$, $cc_7 = \{3\}$,

that are mapped to the curvature labels CC0, CC2, CC0, CC2, CC0, CC2, CC0, CC2, respectively, from the set {CC0, CC1, CC2}.

The length mapping, not shown in the figure, assigns the length label L0 to all eight constructs:

$$<cc_i> \longrightarrow l(cc_i) = L0 \quad (0 <= i < 8\}, i \varepsilon Z \tag{5.6}$$

since all the constructs consist of fewer than 25% of the total number of curvature elements in the whole object, (5.3).

The left and right directional mappings for the curvature construct $<cc_0>$, to neighbouring constructs $<cc_1>$ and $<cc_7>$, respectively, are indicated, in Figure 5.18, by the two black arrows. Again, to avoid clutter, only the numerical component of the directional labels, close to the arrows, is shown. In this example, the direction values refer to left direction D3 and right direction D0, from the direction wheel in Figure 5.10. The full description of these *Level* 3 directional mappings for curvature construct $<cc_0>$ is:

$$<cc_0> \longrightarrow d_{left}(c_{left\ end\ of\ construct}) = \mathbf{D3}_{L3} \quad \varepsilon \{D0_{L3}, D1_{L3}, \ldots\ldots, D7_{L3}\}$$

$$<cc_0> \longrightarrow d_{right}(c_{right\ end\ of\ construct}) = \mathbf{D0}_{L3} \quad \varepsilon \{D0_{L3}, D1_{L3}, \ldots\ldots, D7_{L3}\} \qquad (5.7)$$

Thus, the complete mapping for curvature construct $<cc_0>$ is given by:

$$<cc_0> \longrightarrow (CC0, L0, D3_{L3}, D0_{L3}) \qquad (5.8)$$

5.3.1.2.5 Level 4

Whole objects are formed at *Level* 4 through the application of the relation $R_{3\text{-}4}$, that assembles sets of connected curvature constructs into closed shapes:

$$<<c_0>, <c_1>, \ldots\ldots, <c_m>> = <w_i>$$

with the resulting objects, $<w_i>$, being mapped to the appropriate object class label:

$<w_i> \longrightarrow$ {circle, square}, Figure 5.19.

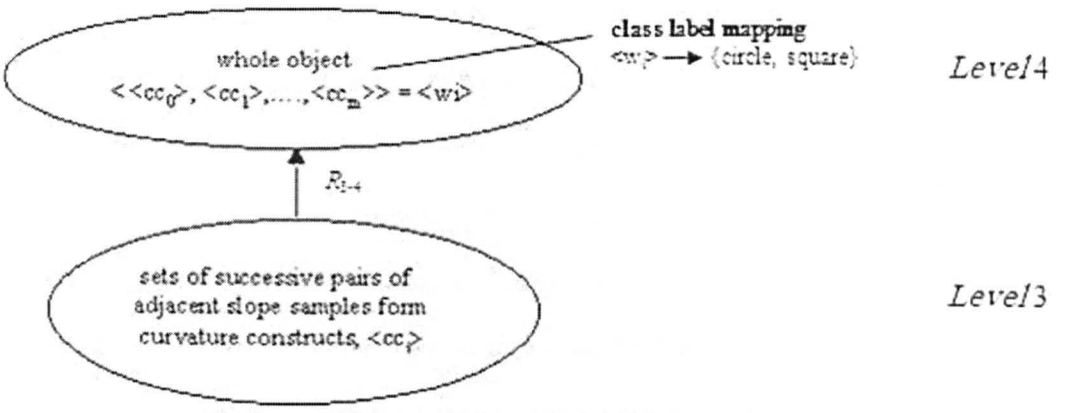


Figure 5.19: Assembling sets of curvature constructs to form whole objects from Figure 5.21

5.3.1.3 Curvature construct encoding and object representation

It can be seen that the representation has been kept very simple, with the multilevel processing designed to give rise to constructs at Level 3 that are characterised by just four attributes - curvature, length-ratio, and left and right generalised directions, (5.8). Also, the range of values these parameters can take has been considerably curtailed. These restrictions are designed to keep combinatorial explosion under control and to allow the system to generalise.

Thus, a whole object, such as the square in Figure 5.18 can be described by the ordered set of encodings of its constituent curvature constructs, Figure 5.20.

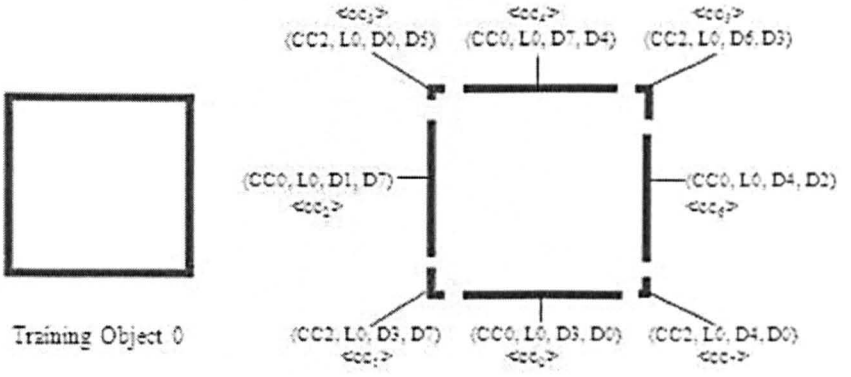


Figure 5.20: Representation of training object 0 as a set of eight coded curvature constructs

The structure in Figure 5.18 fits together uniquely in a configuration in which a construct, $<cc_i>$, can connect to a neighbour on its right, say $<cc_{i-1}>$, if and only if the difference between the numerical parts of the *right* direction of construct $<cc_i>$ and the *left* direction of construct $<cc_{i-1}>$ is equal to four, since this difference implies opposite directions in the generalised direction wheel, Figure 5.10. Similarly, construct $<cc_i>$ can connect to a left neighbour, $<cc_{i+1}>$, if and only if the numerical parts of the left direction of $<cc_i>$ and the right direction of $<cc_{i+1}>$ differ by four. This is referred to as the 'rule for connectability' and can be seen in Figure 5.20 taking constructs $<cc_{i+1}>$ and $<cc_{i-1}>$ as the left and right neighbours respectively of construct $<cc_i>$, $(0 <= i < 8)$. A summary of the entire multilevel process - the pre-processing phase and the representation/recognition phase - can be seen in Figure 5.21.

Figure 5.21: Multilevel architecture for representation and recognition of visual objects

5.3.2 Second set of experiments - Phase2: Representing structure with hypernetworks

5.3.2.1 Multidimensional relations and shared structure

As discussed in Chapter 4, Section 4.4.1, *hypernetworks* allow generalisation from the binary or pairwise relations, depicted by the vertices and edges of networks, to multidimensional relations represented by the vertices and edges of interconnected sets of simplices. A set of simplices is called a *simplicial family* (Johnson, 2006), and a *hypernetwork* is defined to be a *simplicial family* with all its intersecting faces (Johnson, 2007). An individual simplex representing the relationship among *n* things can be depicted as a polyhedron with n vertices in an (*n*-1)-dimensional space, as seen in Figure 4.10, p150.

At every processing level in Figure 5.21, a given structure can be represented by a simplex, the vertices of which correspond to the elements at the previous level, say *N*-1, that are brought together under a relation, or relations, to form the simplex at level *N*. For example, a set of pixels at preprocessing Level 0, having been assembled under the relation of 'darkness' and horizontal contiguity to form a 'dark run', can be represented as the vertices of a 'dark run' simplex at preprocessing Level 1. If the number of pixels constituting the dark run is m, then the resulting simplex has m vertices and $m - 1$ dimensions and can be referred to as an $(m - 1)$-simplex. In the same way, an object at Level 4 of Figure 5.21 can be represented by a simplex, the vertices of which represent its constituent curvature constructs at Level 3. A specific example is illustrated in Figure 5.22. Here, a circle, part (c) of the figure, comprised of the four curvature constructs in (b), is represented by a tetrahedron or 3-simplex, part (a). Each vertex has been labelled with the curvature code of the curvature construct it represents and the edges between the vertices associated with pairs of adjacent connected constructs have been assigned their right and left directional codes, shown in grey. Again, to reduce clutter, only the numerical part of the codes has been used and the length-ratio values have been omitted in part (a).

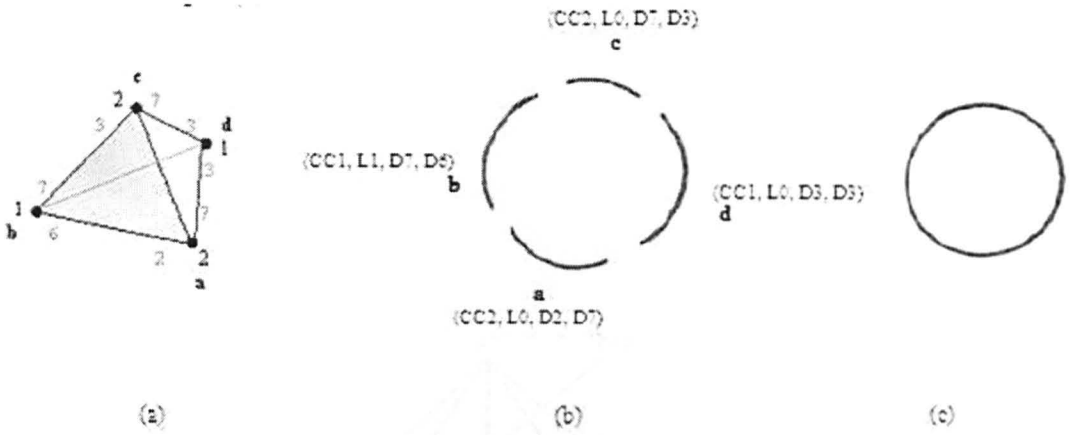
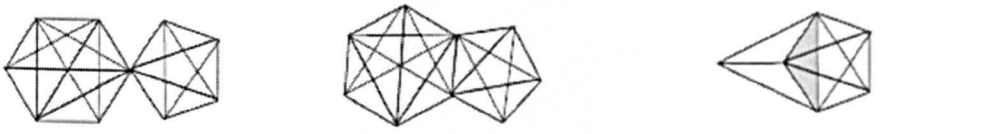


Figure 5.22: Shape representation using a polyhedron

Note the difference between directional values on each edge between connected constructs is 4, in keeping with the 'rule for connectability'. The 'exploded' version of the circle in part (b) shows the structures that give rise to the four constructs represented by the tetrahedron, with their full labels.

When pairs of shapes have some structure in common, this can be depicted in the sharing of vertices within a simplicial complex. They are said to be 'q'-near, where q is the number of dimensions of the shared polyhedral face (Johnson, 2007). For example, two shapes might have one, two, three or more shared constructs, and thus be 0-, 1-, 2- or 'more'-near, respectively, Figure 5.23.

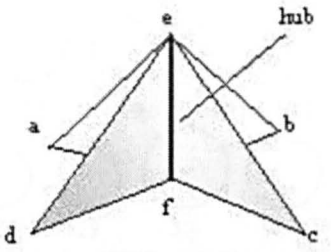


(a) 1 shared vertex (0-near) (b) 2 shared vertices (1-near) (c) 3 shared vertices (2-near)

Figure 5.23: Simplices connected at different dimensions
from Johnson 2005

When multiple shapes share structure, their simplicial representation forms a star-hub configuration (Johnson, 2007). This common structure is the intersection of the sets of constructs comprising the related objects, and can be represented as a hub, with the associated

objects forming a surrounding star of simplices. Figure 5.24 duplicates Figure 4.11 to show a 1-dimensional hub and its star, depicting a pair of constructs, <e> and <f>, shared by the four shapes, <a, e, f>, <b, e, f>, <c, e, f> and <d, e, f>.



The simplices, ⟨a, e, f⟩, ⟨b, e, f⟩, ⟨c, e, f⟩ and ⟨d, e, f⟩ share the face ⟨e, f⟩

Figure 5.24: A star-hub configuration
after Johnson, 2006, Figure 14

A number of the training shapes used in this current work, Figure 5.25, have been found to have several curvature constructs in common. Illustrating such multidimensional connectivity diagrammatically can be difficult, so an alternative representation is used here, in the form of an incidence matrix, Table 5.1, to facilitate analysis of shared structure.

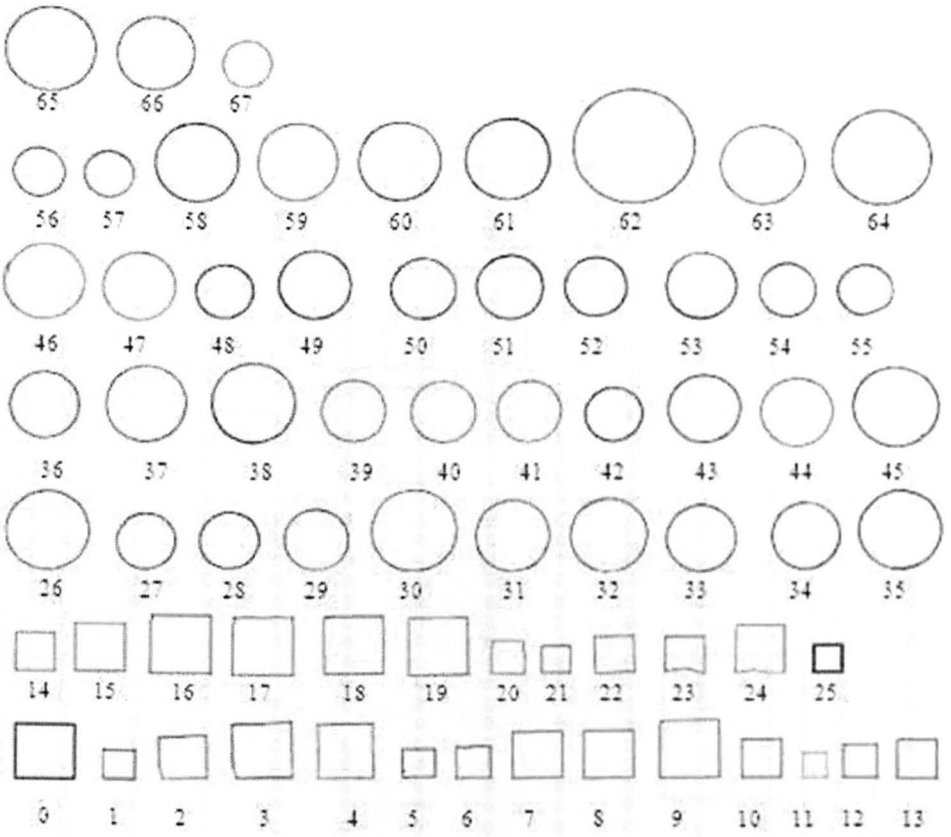


Figure 5.25: The circles and squares of the training set

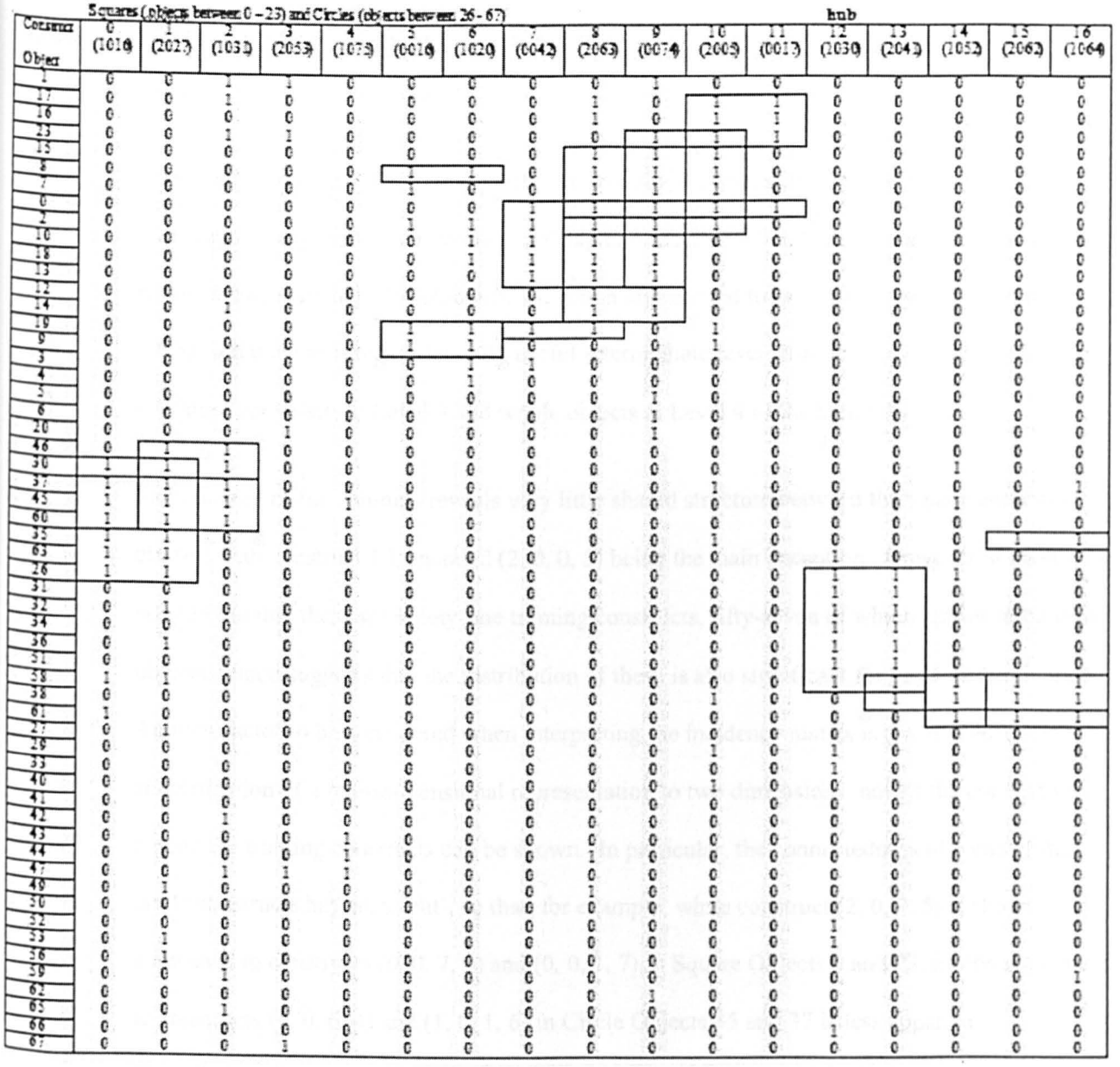


Table 5.1: Incidence matrix of hub-constructs and associated circles and squares

Across the top of Table 5.1, seventeen of the most frequently occurring curvature constructs in the training set are shown. Each construct is connected to its immediate neighbours according to the 'rule for connectability', Section 5.3.1.3. Again, only the numerical components of the construct codes are shown. Down the left hand side of the Table are listed the training objects in Figure 5.25 that include one or more of these constructs in their description. Each object is represented as a row the matrix, with a '1' indicating the occurrence of a particular construct. For example, Object 1 has constructs 2, 3 and 9. Horizontally adjacent 1s show connected structure within an object, while several 1s appearing in a column indicate that a particular

construct belongs to multiple objects. Objects in the range 0 – 23 are squares and the remaining objects are circles.

Connected structure belonging to more than one object is made apparent through appropriate ordering of the objects. As explained in Chapter 4, Section 4.5.3, the associated regions appear in the matrix as rectangular blocks of 1s, which are referred to as *maximal rectangles* (Johnson, 2006), and they represent potentially useful intermediate level structure between the individual curvature constructs at Level 3 and whole objects at Level 4 of the hierarchy.

Examination of the columns reveals very little shared structure between the square and circle classes, with construct 10, encoded (2, 0, 0, 5) being the main exception. However, it should be pointed out that there are ninety-one training constructs, fifty-seven of which appear in multiple objects, which suggests that the distribution of these is also significant for good discrimination. Another factor to be considered when interpreting the incidence matrix is that, because it is a simplification of a multidimensional representation to two dimensions, not all the connectivity among the training constructs can be shown. In particular, the connectedness of some of the circle constructs has been 'cut', so that, for example, while construct (2, 0, 0, 5) is shown connected to constructs (0, 0, 7, 4) and (0, 0, 1, 7) in Square Objects 0 and 23, its connectedness to constructs (1, 0, 6, 4) and (1, 0, 1, 6) in Circle Objects 35 and 37 is less apparent. Nevertheless, it can be seen from the table that the overall distribution of the multilevel structure shows good clustering of the classes with respect to the seventeen selected constructs. In addition, the table shows that when a mixed-category construct such as (2, 0, 0, 5) is combined with a more class-specific construct like (0, 0, 7, 4), the resulting joint structure becomes entirely specific for one class, in this case, the squares.

5.3.2.2 Results and Analysis of Phase 2 of second set of experiments

The connectedness of the multilevel hypernetwork representation provides a 'backcloth' for the image-processing 'traffic' of the mappings defined at each level of the system in Figure 5.21. Thus the mappings at each level are constrained by the topology of the connected simplices (Johnson, 2007). For example, at Level 1, the connectivity of the simplices representing each of the 169 light and dark pixels of a 13x13 slope sample, directs the mapping of that slope sample to a particular slope element, while at Level 4, the connectivity of the simplices, the vertices of which represent the curvature constructs comprising whole objects, causes incoming structure to be mapped to one of the two 'object' classes known to the system – 'circle' or 'square'.

Therefore, if object recognition is taking place at Level 4, where whole objects are represented, the implication is that the approach to recognition is to match the set of curvature constructs from an incoming test object to a set of constructs constituting a known object in the database. On the other hand, if the system were to make use of information about the connectivity and class-dependent characteristics of structure at the curvature construct level and above, but excluding the 'whole object' level, then a sufficiently confident classification might be made on the basis of this intermediate-level structural knowledge. Thus the experiments described in this section can be used to:

Point (1) show that this multilevel hypernetwork approach can be used to represent simple shapes and to discriminate them at the 'whole-object' level, through a simple process of curvature construct matching based on the assumption that sufficient spatial information is encapsulated in the construct mappings at Level 3, Figure 5.21, to eliminate ambiguity in their configuration at Level 4.

Point (2) explore the possibility that knowledge of the nature and connectedness of intermediate-level structure - in the sense of knowing whether subsets of constructs, connected in accordance with the 'rule for connectability', tend to occur in one class of shape rather than another – can enhance recognition confidence.

5.3.3 Second set of experiments – Phase 3: Matching objects at different representational levels

5.3.3.1 'Whole-object' matching

In the first trial, previously unseen objects are matched against whole objects in the database. The curvature constructs comprising an incoming object are compared one-by-one with the members of the set of constructs in each training object. The numerical values representing the curvature type, length ratio and the left and right generalized directions are compared and a test construct must match a training construct exactly to be counted towards the recognition score for a given object. A test object is assigned to the class of the training object with which it shares the highest number of constructs. However, the matched objects are not required to have the same number of constructs and the test constructs do not necessarily have to occur in the same sequence as their matches in the training object. In other words, recognition is not dependent on the precise alignment of the simplices representing the training and test objects in question. Instead, the degree of recognition or strength of the mapping at Level 4 is dependent on the 'q-nearness' of the training- and test-simplex pair, Figure 5.26.

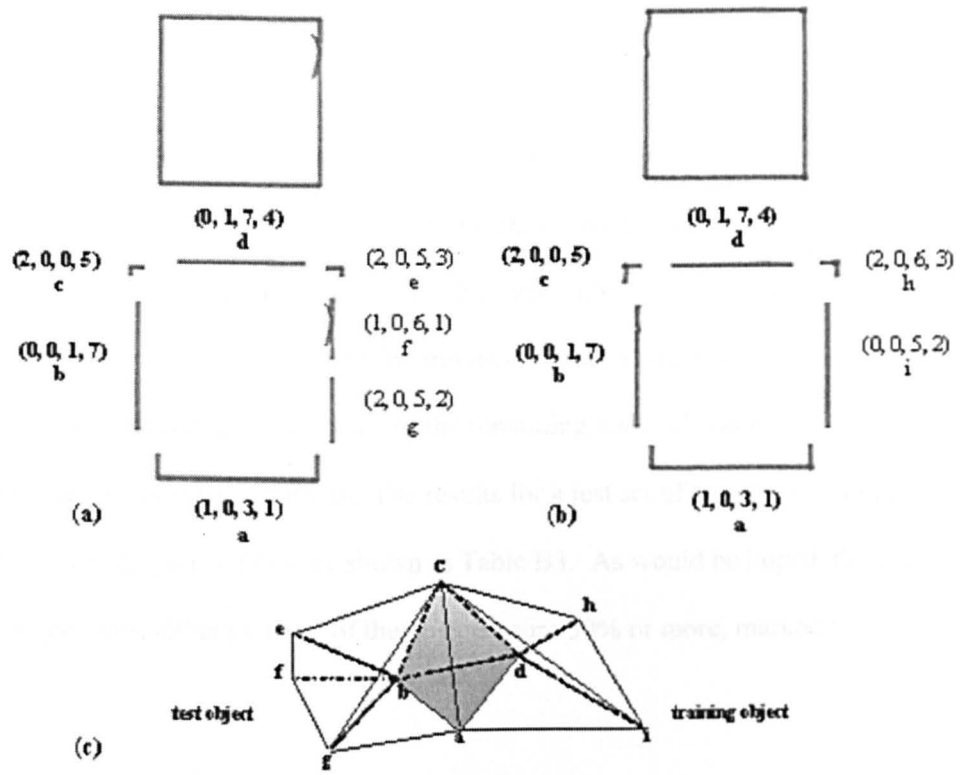


Figure 5.26: Comparing objects in terms of their q-nearness
(a) Test Object with 7 constructs. (b) Training Object with 6 constructs. Highlighted constructs 'a', 'b', 'c' and 'd' are common to both shapes. (c) Polyhedral representation of the shape simplices with the shaded tetrahedron showing shared structure. These shapes are 3-near.

This examines the assumption in Point 1 above, that the relational mappings of length-ratio and generalised direction that are built in to the representation of individual constructs at Level 3 reflect a spatial configuration of a set of constructs of appropriate curvature type that is characteristic of the class of the given input shape.

The overall match score for a test object is expressed as a percentage of the ratio of correctly matched constructs to the total number of constructs in either the training object or the test object, whichever has the larger number of constructs, (5.9).

Match score = 100*number-of-matched-test-constructs/total-number-of-constructs (5.9)

Thus, only a complete match between a test object and a training object with the same number of constructs can score 100% match.

The results of this trial are shown in Tables B1 – B3 (Appendix B). In each table, the number of training constructs and test constructs are shown, along with the number of matched constructs,

the percentage match and the number of the matched training object. Training squares are numbered from 0 to 25 and circles from 26 to 67. Table B1 shows that, out of a set of fifty previously unseeen squares (Figure 5.27), forty-five of them are 100% correctly classified and that four of the remaining five squares are classified as a square with at least 50% confidence, while one is 'misclassified' as a circle, but with only 25% confidence – marked '!'. Table B2 has a similar result for a set of fifty previously unseen circles (Figure 5.28), with forty-four 100%-correct recognitions. None of the remaining six circles is recognized better than 50%, but there are no misclassifications. The results for a test set of twenty-five objects – ellipses and polygons, (Figure 5.29) – are shown in Table B3. As would be hoped, there are no 100% recognitions, although three of the objects score 50% or more, marked '^'.

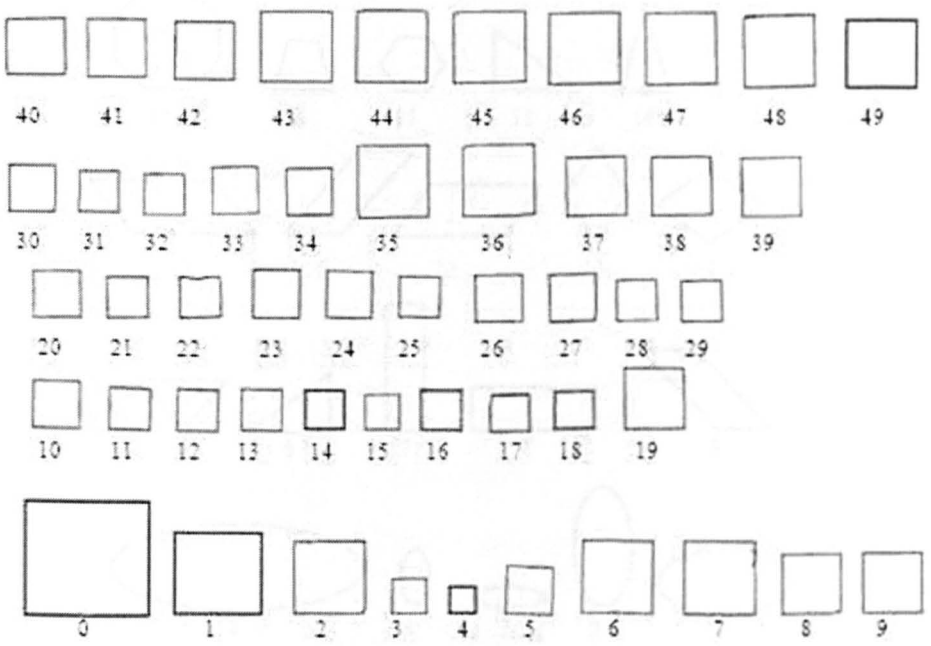


Figure 5.27: Test squares

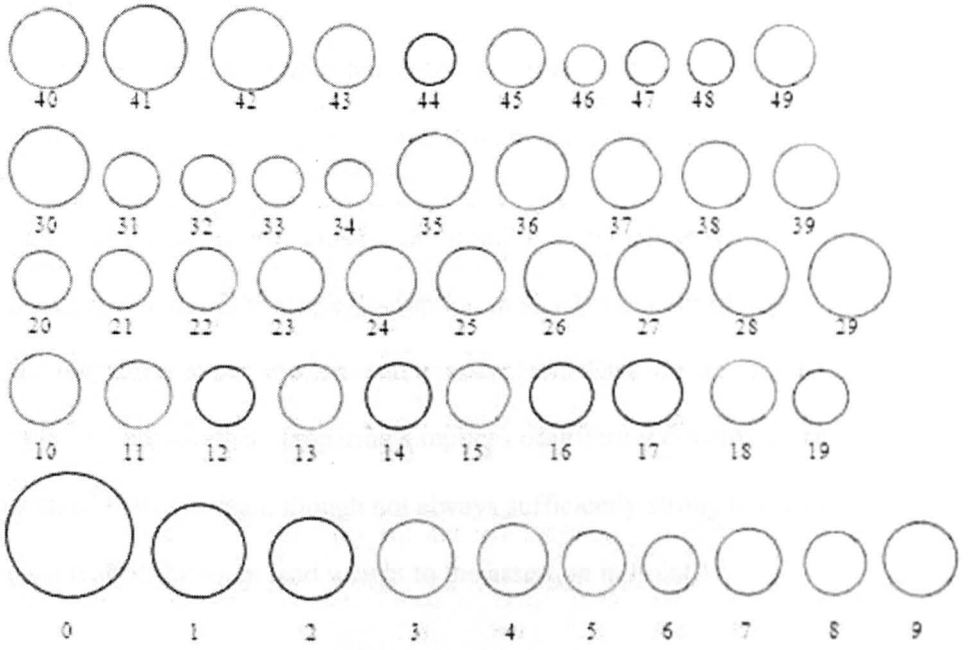


Figure 5.28: Test circles

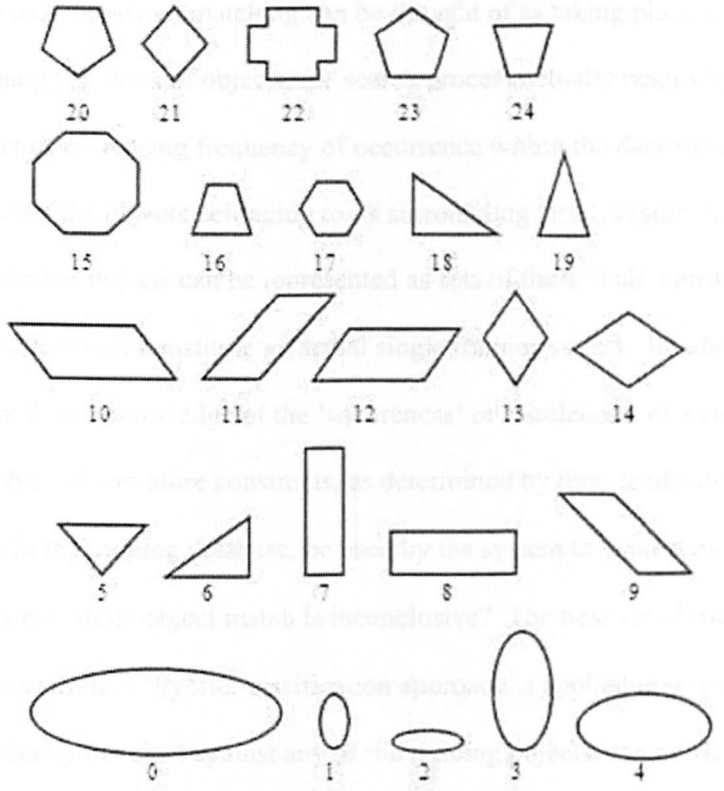


Figure 5.29: Polygons and ellipses are discriminated from circles and squares

Thus, if a classification threshold is set at 80%, so that none of the squares and circles is misclassified, 92% of the squares and 88% of the circles are correctly recognized, while none of the ellipses and polygons scores highly enough to be misclassified as a circle or a square.

Although the scale of these experiments is very small, in terms of the size of the training and test sets and the restriction of the shape description to just two classes of object, there is an indication that multilevel hypernetworks can represent and discriminate simple shapes at the whole object level, and also that comparing simplices of differing dimensions can often yield a shape 'hypothesis' that is correct, though not always sufficiently strong to permit reliable classification. Hence the results lend weight to the assertion in Point 1.

5.3.3.2 Recognition using intermediate-level structure

While in the first trial, construct matching can be thought of as taking place within the simplices representing training/test pairs of objects, the search process actually begins by examining 'hub' constructs in order of decreasing frequency of occurrence within the database, a matching 'hub' triggering a search of the objects belonging to its surrounding 'star', Figure 5.24. This leads to the question of whether objects can be represented as sets of these 'hub' constructs, regardless of whether they collectively constitute an actual single training object. In other words, as suggested in Point 2, can knowledge of the 'squareness' or 'circleness' of a curvature construct or a connected subset of curvature constructs, as determined by their tendency to occur in squares or circles in the training database, be used by the system to make a more confident classification, when a whole object match is inconclusive? The next set of trials explores this hypothesis. In these trials, a 'hybrid' clssification approach is applied to test objects that are not sufficiently confidently matched against any of the training objects. As a first step, the constructs in the test object are matched against all the hub constructs in the database and an initial 'hybrid' match score is obtained through the formula:

hybrid-match-score =
100*number-of-matched-constructs/number-of-constructs-in-test-object (5.10)

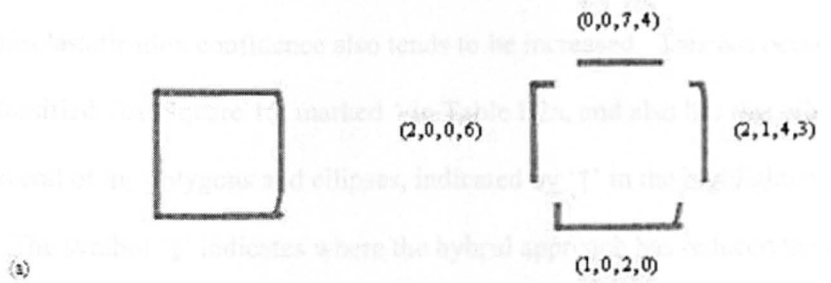
5.3.3.2.1 Hybrid matching 1

In the first of these trials, referred to as 'hybrid matching 1', this result is then modified, according to whether the matched constructs are associated more frequently with circles or squares, by the formula:

$$\text{final score} = \text{hybrid-match-score}*\text{max-matches}/(\text{square-matches} + \text{circle-matches}) \qquad (5.11)$$

where square-matches and circle-matches are the overall number of squares and circles, respectively, associated with all the constructs comprising the test object and max-matches is the larger of these two values.

For example, Figure 5.30(a) shows Test Square 30, from Figure 5.27, with four constructs, all of which are matched across the database, giving a hybrid match score of 100% in Table B2a (Appendix B). Again, non-numerical values have been omitted from the encoding. Also in Figure 5.30(b), each of the four hybrid matched constructs is shown with its associated 'simplex' of associated objects. Summing over all four constructs, we find that they appear in a total of thirty of the training squares and four of the training circles. Hence, from (5.11), the final score for Test Square 30 is given by $100*30/(30 + 4) \approx 88\%$.



Construct	Object list
(1,0,2,0)	< 2,3,4,6,8,9,10,18,19 >
(2,0,0,6)	< 3,4,6,10,20,67 >
(0,0,7,4)	< 0,1,2,5,6,7,8,10,12,13,14,15,18,20,23,62 >
(2,1,4,3)	< 5,52,53 >

(b) Squares numbered 0 – 25, circles numbered ≥ 26

Figure 5.30: Hybrid-matching of a test square
(a) Test Square 30 with its four 'hybrid-matched' constructs
(b) The four constructs and their associated object lists

Tables B2a, B2b and B2c, in Appendix B, compare the results for the 'whole object' matching scheme, shown on the left of each table, and this first hybrid approach, shown on the right. The right hand portion of the three tables gives the overall percentage of test constructs matched, the number of square construct matches and circle construct matches and the overall hybrid percentage square score or circle score, whichever is the higher.

Table B2a compares the results for the five test squares that were not 100% correctly recognized under the 'whole object' scheme. Table B2b gives the equivalent results for the five test circles with a non-zero 'whole object' score of less than 100%. Comparing the highlighted columns on each side of Table B2b, we see that under the 'hybrid matching 1' scheme, all the circle scores are increased considerably, with one score, marked '*' being elevated above the 80% threshold. However, with the squares the result is more mixed. The highlighted columns in Table B2a show two of the squares are now 'misclassified' instead of only one – indicated by '!', although the scores of 48% and 50% are well below the recognition threshold. A third square has its score raised above the threshold – marked by '*', while a fourth is reduced to below the threshold – indicated by '^'.

Enhancing the classification confidence for the squares and circles naturally introduces a trade-off in that misclassification confidence also tends to be increased. This has occurred in the case of the misclassified Test Square 15, marked ^ in Table B2a, and also has this effect on the scores of several of the polygons and ellipses, indicated by '↑' in the highlighted columns of Table B2c. The symbol '↓' indicates where the hybrid approach has reduced the recognition score.

5.3.3.2.2 Hybrid matching 2

In the next trial, which will be referred to as 'hybrid matching 2', the assumption is that, under certain conditions, knowledge of whether an individual construct appears exclusively in circles or squares in the database, allows the system to infer the designation of its neighbouring constructs in the test object. The method begins by obtaining the 'hybrid match score' using

(5.10), as before, and then each matched construct is assigned to one of three different categories – 'S', if the construct only appears in squares in the database, 'C' if it only appears in circles, and 'SC' if it appears in both squares and circles. The mixed-class 'SC' constructs can then be converted to a single-category construct 'S' or 'C' according to the following 'construct-conversion' rule. If an 'SC' construct is connected - as in the 'rule for connectability' - on one side, to a single-category construct, it can be converted to that type of construct provided it is not connected to a different single-category construct on the other side. For example, if we have a connected construct sequence {'S', 'SC', 'SC'}, or {'SC', 'SC', 'S'}, the centre 'SC' construct can be converted to an 'S' construct, whereas if we have the sequence {'S', 'SC', 'C'} it cannot, and must remain a category 'SC' construct. Once the permitted category conversions have been carried out, a 'shape coefficient' is calculated. First, the single-category constructs of both types are counted – each contributing two points to their respective 'square' and 'circle' tallies, and then the remaining unconverted mixed-category constructs contribute one point to each class. The 'shape coefficient', A, is then derived by dividing the larger of the 'square' and 'circle' tallies by the maximum possible count – 2*number-of-matched-constructs. The test object is then assigned to the class of the 'shape coefficient', A, with confidence A*I*, where I is the initial hybrid percent score. Figure 5.31 illustrates the process with Test Square 44 from Figure 5.26.

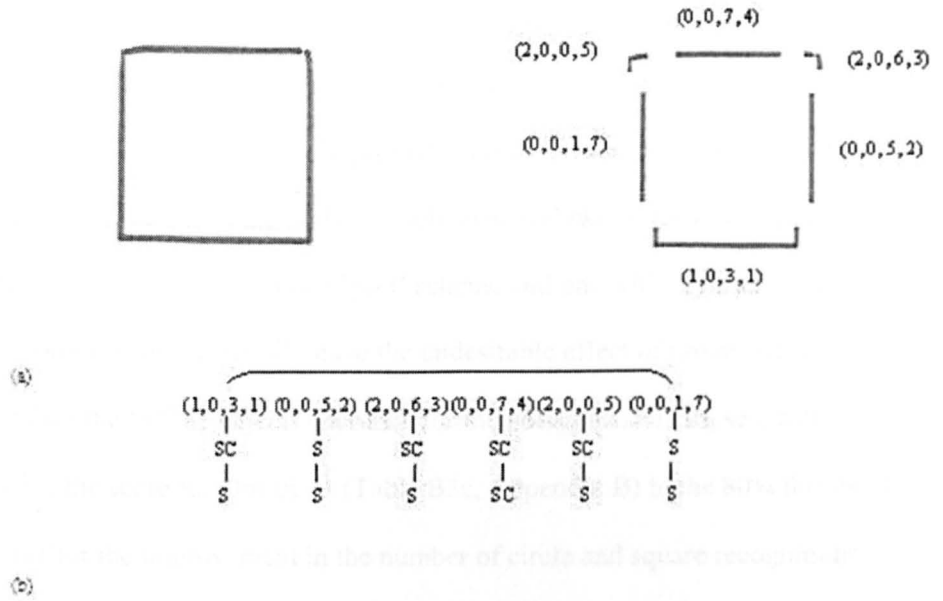


Figure 5.31: Matching a test square with 'hybrid-matching 2'
(a) Test Square 44 with its six 'hybrid-matched' constructs
(b) The six constructs – upper row - and their categorisation as 'square'-, circle-, or 'mixed'-category – middle row. Adjacent constructs are connected according to the 'rule for connectability'. The bracket indicates that the end constructs are also connected. The bottom row shows the permitted 'hybrid matching 2' category conversions from 'mixed' to 'square'.

Tables B3a and B3b (Appendix B) compare the whole-object matching results for squares and circles respectively, on the left, with those for 'hybrid matching 2', on the right. The highlighted columns in Table B3a show that this hybrid scheme increases all the classification scores, taking the score for Test Square 30 above the 80% recognition threshold. The score of the misclassified Test Square 15, is also increased to 50% from 25%, but is still well below the 80% recognition threshold, while Test Square 24, which was misclassified under 'hybrid matching 1' is now correctly assigned again. Scores that are above the threshold are marked by '*', while increased misclassification scores are marked '!'. In Table B3b, the highlighted columns indicate that all the classification scores are increased, with two being raised to 100% confidence. Again the scores for most of the polygons and ellipses are increased, as indicated by '↑' in the highlighted columns of Table B3c, and in addition one of them is taken to the 80% recognition threshold. The symbol '↓' marks a reduction in classification confidence, as before.

Comparing the two hybrid schemes and the 'whole object' scheme discussed so far, it can be seen that with 'hybrid matching 2', recognition performance with the circles and squares is generally better than with the 'whole object' and 'hybrid matching 1' schemes. The classification scores are mostly improved. Two of the squares score above the 80% threshold as opposed to just one with the other two schemes, and two of the circles score above the threshold instead of none with the 'whole object' scheme and one with 'hybrid matching 1'. However, both hybrid schemes generally have the undesirable effect of producing higher classification scores than the 'whole object' approach for the polygons and ellipses, with 'hybrid matching 2' raising the score for Object 24 (Table B3c, Appendix B) to the 80% threshold, an unfortunate trade-off for the improvement in the number of circle and square recognitions.

5.3.3.2.3 Hybrid matching 3

The third hybrid scheme, referred to as 'hybrid matching 3' requires that a mixed-category construct that is suitably connected to single-category construct within the test object, as specified in 'hybrid matching 2', can only be converted to the category of its neighbour if the same pairing occurs in at least one object in the training set, Figure 5.32. Otherwise, the scoring method is the same as that of 'hybrid matching 2'.

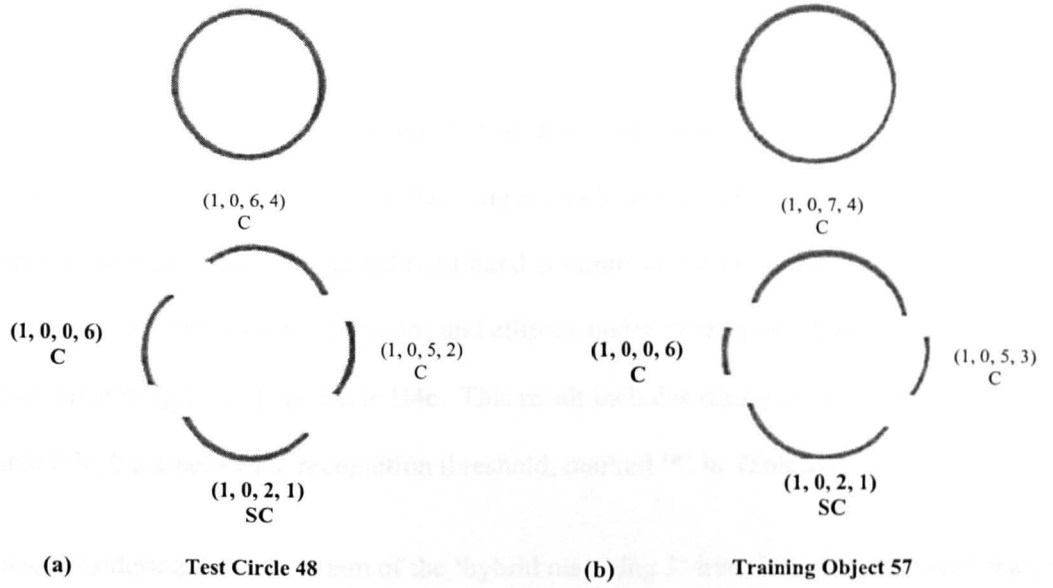


Figure 5.32: Applying the 'hybrid-matching 3' rule
Applicaton of 'hybrid matching 3' rule to convert construct (1, 0, 2, 1) from category 'SC' to category 'C'. Construct (1, 0, 2, 1) appears in Test Circle 48 in (a), connected on each side to a construct of category 'C', which is sufficient to satisfy the category conversion rule of 'hybrid matching 2'. In addition, the connected construct pair, (1, 0, 2, 1) and (1, 0, 0, 6) in Test Circle 48, also appears in Training Object 57, which fulfils the additional requirement of 'hybrid matching 3'.

The aims of this trial are:

i) to assess the validity of the assumption, in 'hybrid matching 2', that intermediate-level test object structure, in the form of appropriately-connected single-category-construct/mixed-category-construct pairings, provides reliable classification information.

ii) to attempt to reduce the undesired increase in classification confidence of 'hybrid matching 2' with the polygons and ellipses, while at least maintaining its recognition performance with the circles and squares.

The results show that 'hybrid matching 3' also raises the 'whole object' recognition scores for the squares and circles, as shown in Tables B4a and B4b (Appendix B), respectively. In addition, comparison of the 'hybrid matching 2' and 'hybrid matching 3' schemes for the squares, highlighted on the right hand side of Tables B3a and B4a, respectively, shows that 'hybrid matching 3' lowers one score – '↓', raises another - '↑', and maintains two scores above the 80% threshold, '*', in Table B4a. The circles do not fare quite so well. Tables B3b and B4b

show that 'hybrid matching 3' reduces one 100% score to 75% - ↓, which leaves just one circle score above the 80% threshold – '*', in Table B4b.

As shown in Table B4c (Appendix B), the polygons and ellipses test set 'whole object' scores are also generally raised by 'hybrid matching 3', with one exception, marked '*' in the table. In addition, examining the highlighted right-hand columns of Tables B3c and B4c, it can be seen that four of the scores for the polygons and ellipses under 'hybrid matching 2' are reduced by 'hybrid matching 3', - '↓' in Table B4c. This result includes taking the score of Object 24, in Table B3c, back below the recognition threshold, marked '*' in Table B4c.

Thus, considering the second aim of the 'hybrid matching 3' trial first, it can be seen that, while there is a slight reduction in recognition performance with the circles and squares, some of the desired decrease in classification confidence with the polygons and ellipses is also achieved. This, however, has implications for the reliability of the assumption in the first aim. The slight reduction in the recognition scores occurs because some of the single-category/mixed-category construct pairings in the test objects do not occur in any of the training objects, which makes classification on the basis of the information inherent in this structure potentially risky. This is borne out by the misclassification of Object 24 in the polygons and ellipses test set.

5.3.3.3 Summary of Phase 3 of second set of experiments

All three hybrid schemes increase several of the scores, especially those of Objects 0, 8 and 24 in the polygons and ellipses set, to a rather high level, indicating limitations in the current approach. For example, one problem, illustrated by Object 0 in Table B4c, is that an increased number of recognized constructs – six with the hybrid schemes as compared with just four with the 'whole object' approach in this example – can strengthen the classification bias, especially when the matched structures are all single-category constructs.

Another potential weakness is that no account is taken of the connectedness of opposing classes of single-category constructs. For example, in Object 15, from the polygons and ellipses test

set, Figure 5.29, single-category circle- and square-constructs are connected together, Figure 5.33, which should sound an alarm for any circle or square classification.

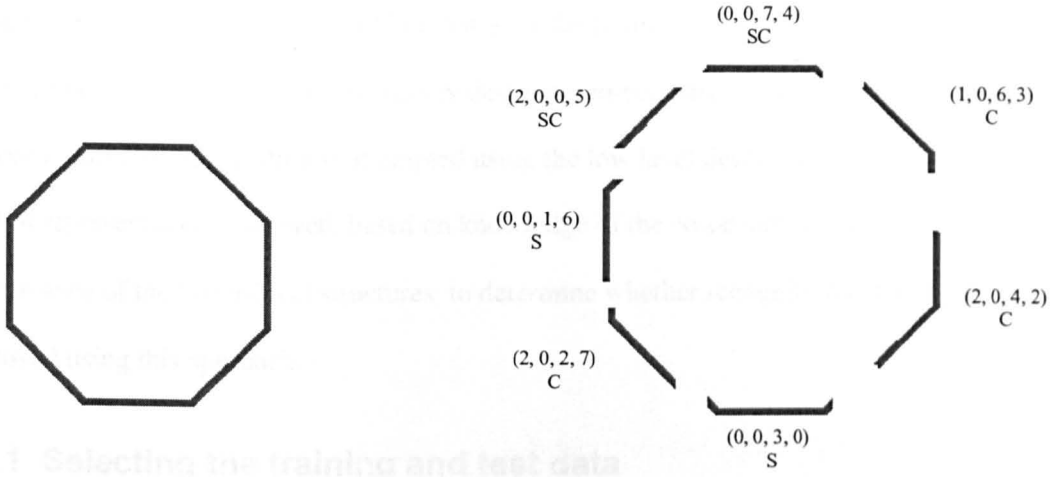


Figure 5.33: Hybrid classification of a polygon
Object 15 from the polygons and ellipses test set shows that when a shape description includes connected subsets of single-category constructs of different classes, as with constructs (2, 0, 4, 2), (0, 0, 3, 0) and (2, 0, 2, 7) in the figure, a circle or square classification is unlikely to be correct.

Hence, the main shortcoming of all three hybrid schemes is that, while in general, they tend to increase the scores of squares and circles, they also increase the circle/square match scores of other shapes. Nevertheless, there is an indication that information about intermediate-level structure could be employed to enhance recognition of objects of known class, lending weight to the hypothesis in Point 2. However, to be sufficiently powerful, any such scheme is likely to have to extend its structural analysis to include examination of the relations between neighbouring single-category constructs as well as those between single-category and mixed-category constructs. Table B5 (Appendix B) summarises the overall performance of the four recognition schemes with the 'squares' , 'circles', and 'polygons and ellipses' test sets.

Overall there is support for the assumptions in Points 1 and 2, with the whole-object and hybrid schemes showing that multilevel hypernetworks can be used to represent and recognize simple shapes, through the matching of structure at different levels. In this small-scale study, the potential effectiveness of individual construct matching in whole objects and matching of construct-pairs within a hybrid representation has been demonstrated.

5.4 Third set of experiments: Object representation and recognition using polygonal descriptions of local image regions

In the third set of experiments, the effectiveness of local low-level polygonal descriptions for representing 'real-world' data in a binary pedestrian/non-pedestrian classification problem is explored. Initially, recognition is attempted using the low level descriptions, and then a higher level of representation is derived, based on knowledge of the co-occurrence and classification performance of the lower-level structures, to determine whether recognition performance can be improved using this approach.

5.4.1 Selecting the training and test data

As described in Chapter 4, Section 4.2.2, the pedestrian and non-pedestrian images used in these experiments are from the Daimler-Chrysler dataset provided by NiSIS for its 2007 pedestrian recognition competition, described in Munder and Gavrila (2006).

It was found, on displaying the various sets of 18x36 pixel images that a sizeable proportion of them had 'blank', grey areas, some quite large, so it was decided not to include any of these in the training or test sets selected for this work, as they would interfere with the task of finding useful image structure for discriminating between pedestrians and non-pedestrians. Thus reduced sets of 100 training images – 50 pedestrians and 50 non-pedestrians, and 200 test images – 100 each of pedestrians and non-pedestrians, were chosen.

5.4.2 The feature extraction and encoding process

The image sampling process is dense, with a 9x9 pixel window being shifted through 280 overlapping locations within the 18x36 images, starting at the lower left corner and, row-by-row, being moved horizontally by one pixel at a time to yield a grid of 10x28 sampling locations, as shown in Figure 5.34.

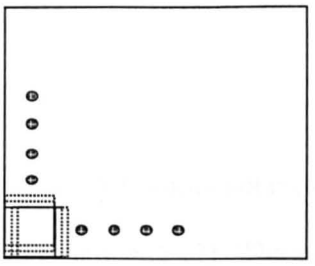


Figure 5.34: The first three sampling windows in the first row and column of the 10x28 grid

Within each window, the mean greyscale value was taken as a threshold and pixels with a greyscale value below the threshold were classed as 'dark' and those with a greyscale value above the threshold were classed as 'light'. These dark and light pixels are then assembled to form dark and light polygons. So each image can be described in terms of 280 overlapping regions containing a number of dark and light polygons and window *j* in image *i* consists of *n* polygons, *m* of which are dark polygons.

Each of the light and dark polygons within a window is described in terms of four 'features':

i) the generalized direction, Figure 5.10, from its centre of mass to the centre of the window
ii) the variance in its horizontal distribution about the centre of mass – 'x'-variance
iii) the variance in its vertical distribution about the centre of mass - 'y'-variance
iv) the greyscale variance

The x- and y-variances and the greyscale variance are put through a 'squashing function' to enable a maximum and minimum value to be known for each 'feature', which is necessary for the implementation of the Relief algorithm (Kira and Rendell, 1992), as explained in Section 5.6.3, below.

The function for the greyscale variance is

$$\text{Activation} = 2/(1 + \exp(-0.1*\text{sdGrey})) - 1 \qquad (5.12)$$

Where sdGrey is the standard deviation of the greyscale values

The 0.1 coefficient is selected because it gives a good spread of the outputs within the range 0-1, and the standard deviation is used rather than the variance to reduce the range of the 'sdGrey' parameter.

The function for the x- and y- variances is

$$\text{Activation} = 2/(1 + \exp(-0.05*\text{variance})) - 1 \qquad (5.13)$$

The 0.1 coefficient of equation (5.12) does not provide such a good distribution of the output values in the range 0 – 1 in equation (5.13). Through 'trial and error' the coefficient of 0.05 is found to give a better spread of the outputs.

Each window is then described by a vector of variable length, depending on the constituent number of polygons found at its particular location in different images. A vector contains information about the number, *n*, of polygons the window contains, the number, m, of these that are 'dark' polygons followed by *n* sets of instantiations of the four features described above.

5.4.3 Learning a 'useful' representation

Much of the information contained in the 280 windows is unlikely to provide for reliable discrimination of the pedestrian and non-pedestrian classes, and so it is important to eliminate irrelevant and 'noisy' descriptions and select only the windows, or, more precisely, the window locations, that provide good discrimination. To this end, a variant of the '*Relief*' algorithm, namely the '*ReliefF*' algorithm (Robnik-Sikonia and Kononenko, 2003), is applied to rank the windows according to their discriminatory 'usefulness'. This variant is explained in Chapter 4, Section 4.3.3 and is repeated in Figure 5.35, below.

Input: for each training instance a vector of attribute values and the class value
Output: the vector W of estimations of the qualities of attributes

1. set all weights $W[A] := 0.0$;
2. **for** $i = 1$ to m **do begin**
3. randomly select an instance R_i;
4. find k nearest hits H_j and k nearest misses M_j;
5. **for** $A := 1$ to a **do**
6. $W[A] := W[A] - \Sigma_{j=1}^{k} \text{diff}(A, R_i, H_j)/(m.k) + \Sigma_{j=1}^{k} \text{diff}(A, R_i, M_j)/(m.k)$;
7. **end**;

Figure 5.35: The modified Relief algorithm used in this study
adapted from Robnik-Sikonia and Kononenko, 2003, Figure 2

The diff() function of line 6 is given below as equation (5.14):

diff(A, I_1, I_2) calculates the difference in the values of an attribute for two instances I_1 and I_2.

$$\text{diff}(A, I_1, I_2) = \frac{|\, value(A, I_1) - value(A, I_2) \,|}{max(A) - min(A)} \qquad (5.14)$$

and is also used in finding the k nearest hits and misses, as described in Section 4.3.3.

The algorithm is adapted slightly for these and the remaining sets of experiments. Because the number of features extracted initially is quite large and *Relief* is based on the Nearest Neighbour algorithm, the technique could suffer the effects of the 'curse of dimensionality', explained in Chapter 3, Section 3.4. Therefore, when the nearest neighbour images are being sought, as in line 4 in Figure 5.35, the difference between image instances is calculated on the basis of the single window under consideration at any given time, rather than all 280 windows. Thus the overall difference between two image instances is the sum of the absolute differences in the values of the n sets of four features for the window currently being evaluated in those images.

5.4.3.1 Application of the modified ReliefF Algorithm

5.4.3.1.1 Finding the *k* nearest neighbours

For these experiments, five nearest neighbours were selected from each class. It was decided to use all the training data in determining the most discriminating windows rather than just a randomly selected subset, so each of the 100 images was taken, in turn, to be R_i. For a given image, R_i, each window, w_j, in turn is compared with a 5-neighbourhood of windows centred on that window's location in all the remaining 99 images, Figure 5.36. To this end, only the 208 non-border windows from the 10x28 sampling grid are examined in each R_i. The matching against a 5-neighbourhood enables the system to generalize the location of similar structure within these variable images, and at the same time, confine the search to within a reasonable area for finding the same type of structure, for example, part of an arm or a leg. This takes advantage of the largely translation- and scale-invariant properties of the dataset.

Figure 5.36: 5-neighbourhood of windows
(a) The grey window represents the position of the window wj currently under examination in image R_i.
(b) shows the actual overlapped relationship of the five windows

Only windows with the same description in terms of the overall number of polygons and the number of dark polygons can be compared. This comparison is carried out across all the 99 remaining images, the diff() function (5.14) being applied between w_j and its eligible 5-neighbours and the five most similar windows from each class being selected for application of *ReliefF*.

5.4.3.1.2 Finding the successful windows

After the algorithm has been applied, the score of window w_j is updated, being increased if the normalized sum of distances between w_j and its five nearest hits, H_j, is smaller than that between w_j and its five nearest misses, M_j, and reduced otherwise, line 6 of Figure 5.35. Once all the images, R_i, have been used, the non-negative total scores for all the 208 windows, w_j, are summed and an average determined. Windows with a score above the average are deemed 'successful' and are selected as the 'features' to represent the training images. 76 such windows were found and the training set representation was adjusted accordingly.

Each window has associated with it a 'star' of training images within which its instantiation is of a particular length. These are the images in which a match will be sought for a corresponding test window.

5.4.3.1.3 Finding the 'useful' images

Although the whole training set was used for learning the most successful windows, some images are likely to have contributed more positively to the result then others, so it was decided that, each time the score of a particular window, w_j, was increased during the application of *Relief*F, the associated image would have its 'usefulness' score incremented. Images with a 'usefulness' score above the average would then be considered eligible to be in the training set. 38 pedestrian images and just 5 non-pedestrian images were above average.

5.4.4 The classification task

Although the pedestrian recognition task this can be thought of as a 2-class discrimination problem, the non-pedestrian class, consisting of anything that is not pedestrian, is obviously a far larger and more variable class than the pedestrian one, so one question is whether a machine vision system should attempt to learn characteristic representations of both classes, or just of the pedestrian class. Another point is that human vision does not need a 'non-class' when learning to recognize new objects.

In this study, the 'successful' windows and 'useful' images were learned through the analysis of both types of image, but for the testing phase, it was decided to explore whether classification performance was better with one class or two in the training set.

For the first trial, only the 38 'useful' pedestrian images were included in the training set. In the second trial, all 50 pedestrian images were used and for the third classification attempt, only the 50 non-pedestrians were included. Finally, the whole training set was used.

5.4.4.1 Classification of new data

As with the training set, it was found that a fair proportion of the test images had a sizeable blank region, so it was decided that for the purposes of this work, only complete images would be selected. 200 such images were chosen, 100 of each class. The training and test images used are shown in Figure 5.37, (a), (b) and (c), respectively.

The broad approach to classification is, for a given test image, to take each 'successful' window from the list of 76, and examine the vector representation of each of its training-image instantiations in turn. For each instantiation, determine the best-matching, compatible window within the corresponding 5-neighbourhood in the test image, using the diff() function in equation (5.14), and either:

i) increment the classification score of the class of the training image that gives rise to the best match, repeating this for all 76 successful windows. Then assign the test image to the class with the larger classification score. Use this approach when the training set is comprised of both pedestrian and non-pedestrian exemplars, or:

ii) when the training set contains just one class, sum all the best-match scores for all of the 76 successful windows and determine their average, then, if the average score exceeds a preset threshold, assign the test image to the class represented by the images in the training set, otherwise, classify it as the opposite class.

The results are summarized in Table 5.2.

Training set	Threshold	Correctly classified pedestrian test images	Correctly classified non-pedestrian test images	Overall score
38 pedestrian images	87.5	76%	95%	85.5%
	87.0	81%	90%	85.5%
	86.5	84%	82%	84.0%
50 pedestrian images	87.5	87%	79%	83.0%
	87.0	92%	71%	81.5%
	86.5	93%	62%	77.5%
50 non-pedestrian images	88.0	82%	38%	60.0%
	87.5	70%	46%	58.0%
	87.0	56%	57%	56.5%
100-strong training set	-	98%	73%	85.5%

Table 5.2: Classification results for the 100 pedestrian and 100 non-pedestrian test images
The highest pedestrian score for each training set is highlighted as are the occurrences of the highest overall score.

The table shows that the best 'balance' in performance between the two classes occurs with the 38 pedestrian images alone and a threshold of 86.5 giving an average score of 84%. The 50 pedestrian images alone significantly improves pedestrian recognition performance, but at the expense of the non-pedestrians. Using the 50 non-pedestrian images alone reduces the outcome for both classes but especially for the non-pedestrians. Finally, incorporating both classes in the

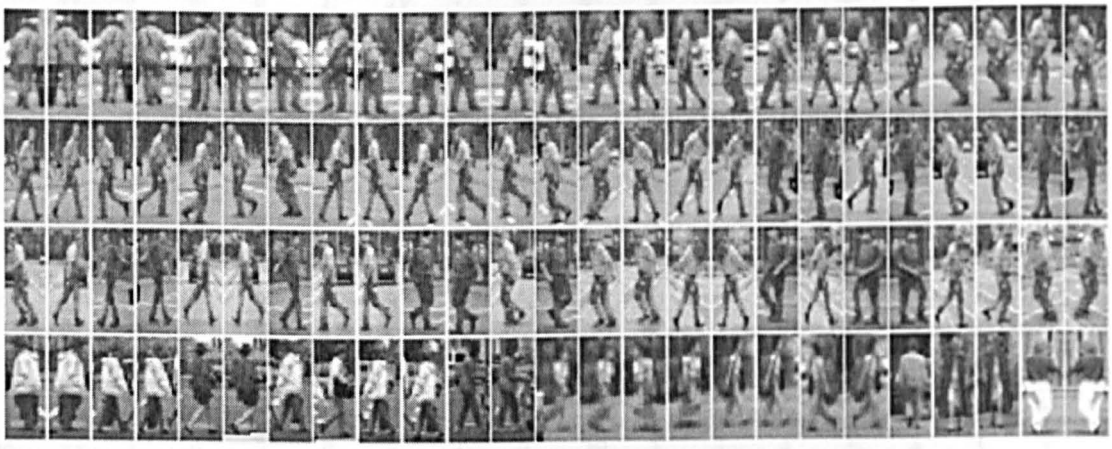
100-strong training set produces the best output for the pedestrians, but with some deterioration for the non-pedestrians.

This suggests that, as with most artificial vision systems, representatives of both classes are required for training and that generally the larger negative class needs more exemplars. In subsequent experiments in this third set, both classes are used for training, however, the issue of determining a suitable quantity of negative training data is not addressed.

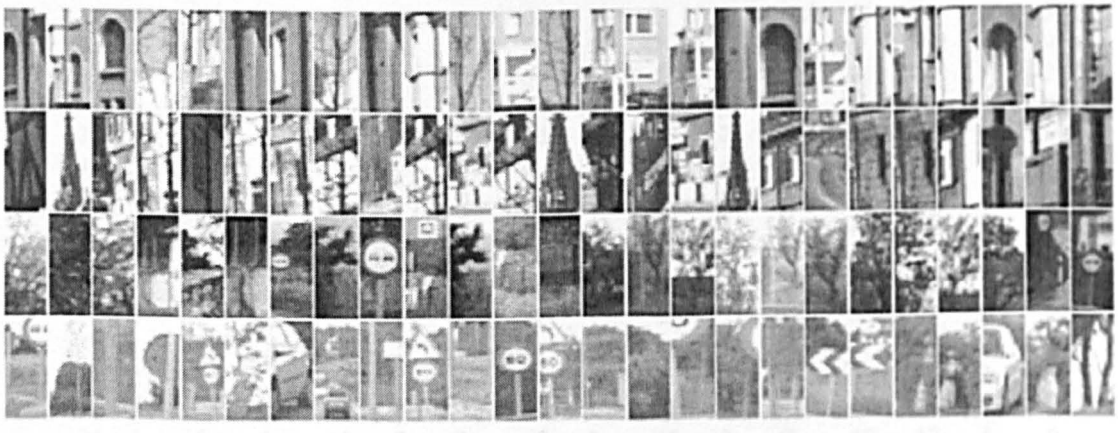
In addition, using just a single class and a threshold assumes linear separability of the two classes, and it can be seen that moving the decision hyperplane by altering the threshold cannot adequately separate the test data, suggesting that a non-linear classification process, such as that provided by a multilevel system is required.

In the next trials, the problem of how higher-level structure might be abstracted autonomously by the system, to resolve ambiguity at the level of the individual windows, is explored.

(a)



(b)



(c)

Figure 5.37: Pedestrian and non-pedestrian training and test sets
(a) the 50 pedestrian- and 50 non-pedestrian training images
(b) the 100 pedestrian test images
(c) the 100 non-pedestrian test images

5.4.5 Learning a 'useful' higher-level representation

The idea is to examine the various instantiations of the 'successful' windows that most frequently discriminate correctly to try to discover if any structural patterns are emerging. To do this, a 76x200 Incidence Matrix, of the 76 training windows against the 200 test images, was constructed, an excerpt of which is shown in Table 5.3.

Training window \ Test image	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	1	8	29	0	0	4	0	0	4	0	3	32	9	32	0
16	2	26	0	0	2	4	48	0	4	48	4	0	0	9	0
17	0	9	0	1	26	4	0	0	0	20	0	0	0	31	0
18	0	7	0	2	7	4	18	0	4	9	9	47	9	14	0
24	1	0	13	33	3	4	15	3	4	3	15	3	20	0	4
31	0	38	0	50	17	0	9	0	9	34	5	3	5	0	5
35	1	22	49	0	41	4	25	3	19	0	19	0	19	41	4
37	23	22	17	0	0	4	6	19	0	44	0	0	0	40	19
44	9	9	50	40	0	1	47	3	4	4	1	3	1	39	5
45	3	9	48	0	15	4	36	3	44	0	0	3	4	3	5
52	13	0	7	8	7	47	19	0	0	0	0	3	25	9	21
53	2	1	2	9	3	47	0	3	4	3	27	3	0	24	2
54	27	9	9	17	3	4	47	3	4	0	27	3	34	27	40
57	45	8	0	14	0	4	0	0	0	0	0	10	0	1	0
58	0	8	40	2	13	2	8	3	4	0	22	3	3	3	0
62	18	0	0	8	21	47	21	21	4	21	0	25	0	5	5
72	25	35	21	0	15	47	0	0	4	21	0	8	44	27	5
73	1	0	7	0	3	47	0	29	4	3	4	3	4	3	5
83	29	1	14	22	7	47	22	0	4	7	22	45	22	45	21
84	1	1	47	47	10	0	0	22	33	0	25	6	25	45	0
85	1	1	47	46	0	10	10	0	4	0	5	0	5	0	0
122	44	1	32	1	28	4	0	10	0	3	44	30	0	3	3
123	0	1	21	1	22	0	1	15	4	0	25	0	0	3	0
132	1	1	14	29	25	0	29	11	1	3	0	0	0	3	3
133	0	1	21	0	0	1	14	0	0	0	0	0	44	3	0
136	1	0	22	0	22	22	46	0	0	0	0	3	5	3	0
143	1	1	0	29	0	38	0	0	4	4	1	0	44	3	0
148	0	0	2	0	0	0	0	22	4	0	26	1	4	1	0
153	1	29	0	0	0	0	0	15	4	3	0	2	15	3	18
154	29	1	0	0	0	33	30	0	0	3	0	3	0	11	21
163	2	27	29	2	0	40	48	15	1	4	4	3	15	3	0
164	1	1	29	0	27	0	1	15	0	22	4	3	0	3	46
173	1	15	29	2	7	28	0	0	15	10	47	3	47	3	48
174	2	2	2	0	48	0	0	49	3	0	4	0	49	3	17
183	44	27	0	49	0	30	27	0	15	28	4	3	4	3	0
184	1	1	1	17	7	0	48	0	45	3	0	48	19	3	0
185	1	1	1	0	17	0	0	18	3	25	0	25	0	6	11

Table 5.3: Excerpt from the full 76x200 Incidence Matrix Appendix D, Table D1, showing a proportion of the 76 successful training windows against the first 15 pedestrian test images from the 200-strong test set, Figure 5.37(b) and (c). The non-zero entries in the body of the matrix indicate the training image in which a window was instantiated when it correctly classified the corresponding region in a particular test image. Zero entries indicate misclassification or a non-match.

A non-zero entry in the body of the matrix, in cell (x, y), gives the number of the training image that matched most closely the test image number at the head of column x, through the window number at the start of row y. For example, test image 5 is matched through window 17 with training image 26. A zero indicates a misclassification or non-classification.

When several non-zero entries have the same value along a row in the matrix, this indicates that a particular training image instantiation of the given window may be a useful tool in correctly classifying multiple test images. For example, window 53, instantiated in training image 3, correctly classifies test images 5, 8, 10 and 12.

When several non-zero entries are the same down a column, this shows the degree of similarity between the associated training image and the given test image, for example, test image 9 is matched with training image 4 in seventeen of the thirty-seven windows. When multiple matching entries along a row also appear in several other rows, this indicates possible higher-level structure.

However, determining which combinations of windows to consider, in which training image instantiations, and for which test images, is potentially very complex, so it would seem to be essential to place considerable constraints on the permitted combinations of windows and on the size and type of neighbourhood in which they can occur.

5.4.5.1 Multilevel classification using an arbitrarily-selected type of 'higher-level' construct

Before making use of the Incidence Matrix, the classification potential of a very simple type of compound window structure, highly constrained within a small region, was explored. This structure is formed from a pair of windows, termed here '2-neighbours' that overlap either horizontally or vertically, shifted by just one column or row of pixels respectively, and is thus contained within a 5-neighbourhood, Figure 5.36. Any given 5-neighbourhood can thus contain up to four 2-neighbours. The 76 'successful' windows used for classification at Level 1 were then paired off according to the above constraints, yielding eighty 2-neighbours, listed in Figure 5.38. The idea was that these 2-neighbours, if they were required to be matched within a single

training image, might help resolve the ambiguity where, at the first level of processing, the same two windows had been matched individually within training images of opposite classes. At this point, rather than having their underlying structures combined to form a joint polygonal representation, the 2-neighbours were matched individually, as before, but required to produce a best joint match score within a single training image. The matching of 2-neighbours in training and test images is explained in Section 5.4.6.

Thus, in the first trial, if a test image was not sufficiently confidently classified above a certain threshold – set at 60% here - any 2-neighbours that did not agree on their classification were rematched within a single training image and the initial score adjusted according to the new joint classification. Of course, even when pairs of windows agree on the classification at Level 1, they can still misclassify, so the next experiment simply ran the Level 2 rematching on all eighty 2-neighbour pairs and rescored accordingly. The experiments were conducted using the full, 100-strong training set and the 200-strong test set, Figure 5.37.

A problem with the reclassification of images that score less than the threshold, is that the system, as well as correcting errors when attempting to resolve ambiguity, may misclassify at Level 2, images it had previously classified correctly at Level 1. There is therefore a trade-off between the level at which the threshold is set to facilitate correction, and the risk of introducing new errors. In this work, the 60% threshold was found, by experiment, to give overall improvement in classification at Level 2, while keeping new misclassifications with scores above the threshold to a minimum.

The two approaches give very similar results. In the top section of Table 5.4, it can be seen that, when the 60% threshold is applied at Level 1, 87 pedestrians and 39 non-pedestrians are sufficiently confidently classified, ie with a match score of greater than or equal to 60%, and 1 pedestrian and 4 non-pedestrian images are confidently misclassified. The number of pedestrian images that can be reclassified, ie they scored below the threshold, is 12, of which only one is a potentially correctable misclassification, while among the non-pedestrians, a total of 57 images score below the threshold, 24 of those being potentially correctable.

The middle section of the table gives the results of Level 2 classification when only conflicting 2-neighbours are re-matched. There are now 91 pedestrian images confidently classified correctly and 6 correctly classified below the threshold, giving a total of 97% correct classifications, which is slightly down on the 98% achieved at Level 1 without taking the threshold into account:- 87 + 11 in the top section of the table. A similar picture for the pedestrians can be seen in the lowest section, which shows the outcome when all eighty 2-neighbours are re-matched. 92 are correctly classified above the threshold and 5 are classified correctly but below the threshold, again giving a total of 97%.

However, Level 2 works a little better for the non-pedestrians. In the middle section of the table, it can be seen that 53, instead of Level 1's 39, non-pedestrian images are confidently classified and a further 21 are correct but below the threshold, giving a total of 74%, which is slightly up on the Level 1 performance of 72% without application of the threshold:- 39 + 33 in the top section. And in the lowest section of the table, the total Level 2 output for correctly classified non-pedestrian images scoring above and below the threshold is 52 + 24 = 76%, up 4% on the Level 1 performance.

Thus, even given the rather arbitrary nature of the higher-level construct selection, the multilevel system, overall, performs slightly better than when only Level 1 processing is applied, with the approach of rematching all eighty 2-neighbours rather than just the 2-neighbours that disagree on the classification, taking the lead.

Section 5.4.5.2 below investigates the effect on performance of refining the selection of Level 2 structure with the help of classification information contained in the Incidence Matrix.

Level 1	Correct >= 60%	Correct < 60%	Non-class or incorrect < 60%	Incorrect >= 60%
Pedestrians	87	11	1	1
Non-pedestrians	39	33	24	4

Rematching just the 2-neighbours that disagreed on classification at Level 1

Level 2	Correct >= 60%	Correct < 60%	Non-class or incorrect < 60%	Incorrect >= 60%
Pedestrians	91	6	1	2
Non-pedestrians	53	21	14	12

Rematching all 80 2-neighbours

Level 2	Correct >= 60%	Correct < 60%	Non-class or incorrect < 60%	Incorrect >= 60%
Pedestrians	92	5	1	2
Non-pedestrians	52	24	13	11

Table 5.4: Multilevel classification results for the 200-strong test set

{15, 16}, {16.17}, {17,18}, {35, 45}, {44, 45}, {44, 54}, {52, 53}, {52, 62}, {53, 54},
{57, 58}, {62, 72}, {72, 73}, {73, 83}, {83, 84}, {84, 85}, {122, 123}, {122, 132}, {123,133},
{132, 133}, {133, 143}, {143, 153}, {153, 154}, {153, 163}, {154, 164},
{163, 164}, {163, 173}, {164, 174}, {173, 174}, {173, 183}, {174, 184}, {183, 184},
{183, 193}, {184, 185}, {184, 194}, {185, 186}, {185, 195}, {186, 187}, {186, 196},
{187, 197}, {193, 194}, {194, 195}, {194, 204}, {195, 196}, {195, 205}, {196, 197},
{196, 206}, {197, 198}, {204, 205}, {204, 214}, {205, 206}, {213, 214}, {213, 223},
{214, 224}, {218, 228}, {223, 224}, {223, 233}, {224, 225}, {224, 234}, {233, 234},
{233, 243}, {236, 237}, {236, 246}, {237, 247}, {241, 242}, {241, 251}, {242, 243},
{242, 252}, {243, 253}, {245, 246}, {245, 255}, {246, 247}, {247, 248}, {251, 252},
{251, 261}, {252, 253}, {253, 263}, {255, 265}, {263, 264}, {264, 265}, {265, 266}

Figure 5.38: The 80 Level 2, 2-neighbour pairs of windows derived from the 76 'successful windows at Level 1

5.4.5.2 Abstracting 2nd-level structure with the help of the Incidence Matrix

Following on from the preliminary work on '2-neighbours' described in Section 5.4.5.1, the Incidence Matrix was taken into consideration for the next trials. The full Incidence Matrix, for the 76 successful training windows against the 200 test images, can be seen in Appendix D. Unlike in the brief excerpt of the Matrix in Table 5.3, there are no '0'-entries in the body of the table. Instead incorrect classifications are apparent where a test image label exceeds 100 while the corresponding training image output is between 1 and 50, or when a test label is between 1 and 100 and the matched training output exceeds 50. For example, window 15 in the first row

of the table mismatches its associated region of test image 4, a pedestrian image, with training image 62, a non-pedestrian image, and conversely also misclassifies its region of test image 112 - non-pedestrian - as pedestrian, through training image 30.

Initial inspection of the Incidence Matrix shows that the 2-neighbour pairs are worth considering as a Level 2 construct, because they appear to classify the corresponding regions of test images correctly in a significant number of cases. For example, the pair comprised of windows 84 and 85 classifies twenty-nine images correctly – for instance, matching pedestrian test images 1, 2 and 3 with training images 1, 1 and 47 respectively, and matchng non-pedestrian test images 110, 118 and 128 with non-pedestrian training images 57, 70 and 73, respectively. However, the pair makes only three incorrect classifications, matching pedestrian test image 97 to non-pedestrian training image 78, and matching non-pedestrian test images 163 and 198 to pedestrian training images 1 and 6, respectively.

The next trials are designed to explore whether using the Matrix to refine the selection of Level 2 structure gives better results than using all the possible instances of the chosen constructs across all the training examples, as in the trials in Section 5.4.5.1.

The system tries to find potentially useful Level 2 structure in two ways.

Firstly, it scans pairs of rows in the matrix, that correspond to the eighty 2-neighbour pairs listed in Figure 5.38, looking for instances of pairs that match a given test image to a single training image, as demonstrated in the example above. Each single-image instantiation of every 2-neighbour pair has a 'usefulness' score, which is incremented whenever the local classification is correct and decremented when it is wrong.

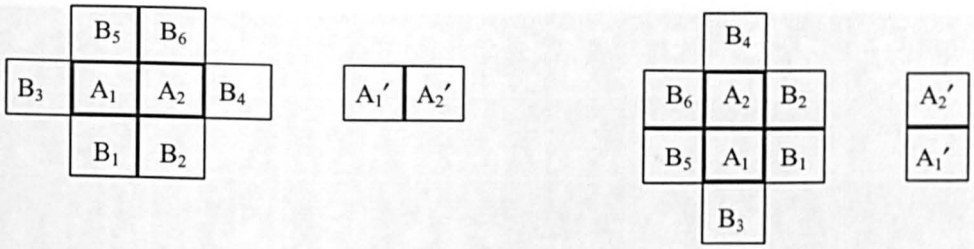
Secondly, it uses the results of re-mapping to a single training image, 2-neighbour pairs that have disagreed on their Level 1 classification – ie, only one window of the pair has classified correctly. First, for the given test image, the region in question is compared with each of the training images that has provided a single-image match with that pair for one or more of the remaining 199 test images. For example, the Matrix shows that the 2-neighbour pair {15, 16}

disagrees on its classification of test image 3, which is matched by window 15 to pedestrian training image 3 and by window 16 to non-pedestrian training image 66. But the pair responds to several other test images with single-image output, some correct and some not. Taking a small sample of the pair's outputs to illustrate, test images 6, 15 and 30 are matched with training images 4, 68 and 6, respectively. So the corresponding region of test image 3 will be compared with images 4, 68 and 6 as well as all the other single-match outputs for the pair. The test region is also compared with each of the two training images involved in the initial classification conflict. In this instance, test image 3 will be matched against training images 3 and 66. The closest match among all the comparisons for that test image then has its 'usefulness' score adjusted according to whether the new classification is correct or not.

Level 2 constructs that end up with a positive score, indicating that they tend to classify correctly more often than incorrectly, are considered 'useful' and their representative vector-pairs are added to the Level 2 data set. Thus, the new Level 2 data contains a substantially reduced set of training vectors, representing a subset of the 100 training examples by a small number of 'useful' 2-neighbour regions, which is variable for each image. The new data consists of 3952 vectors instead of the full 7600.

5.4.6 Multilevel classification revisited

The approach to classification at Level 2 is similar to that at Level 1, Section 5.4.4.1, except that the constructs are 2-neighbour pairs that are now being matched within a compound 5-neighbourhood, Figure 5.39, in each test image.



Compound 5-neighbourhoods and associated 2-neighbour pairs

Figure 5.39: Compound test 5-neighbourhood configurations
The compound test 5-neighbourhood is centred on the equivalent location of a 2-neighbour pair {A1′, A2′} in the training image. The two configurations of 2-neighbours and their associated 5-neighbourhoods are illustrated. The windows are shown without overlap for clarity.

The compound 5-neighbour matching process compares the training 2-neighbours, {A1′, A2′}, with test image window pairs {A1, A2}, {B1, B2}, {B3, A1}, {A2, B4} and {B5, B6}, Figure 5.39. The pair, {A1, A2}, occur in the equivalent location in the test image to windows {A1′, A2′} in the training image.

As at Level 1, only where the lengths of the training and test 2-neighbour vector representations are the same can they be compared. Thus, for example, vectors A1 and A1′ would have to have the same number of elements, as would vectors A2 and A2′.

These trials introduce a new test set composed of 150 each of previously unseen pedestrian and non-pedestrian images, Figure 5.40, which are first classified using the 76 'successful' windows at Level1. As in the previous trials described in Section 5.4.5.1, for the Level 1 classification to be considered sufficiently reliable, the proportion of windows agreeing on the decision must be at least 60%, otherwise the test image in question is reclassified at Level 2. When a test image score is below the threshold, the 2-neighbour pairs, whose members have been matched individually at Level 1 within training images of opposite classes, are rematched against corresponding 2-neighbours within single training images, at Level 2. Only window-pair descriptions of the same length can be compared.

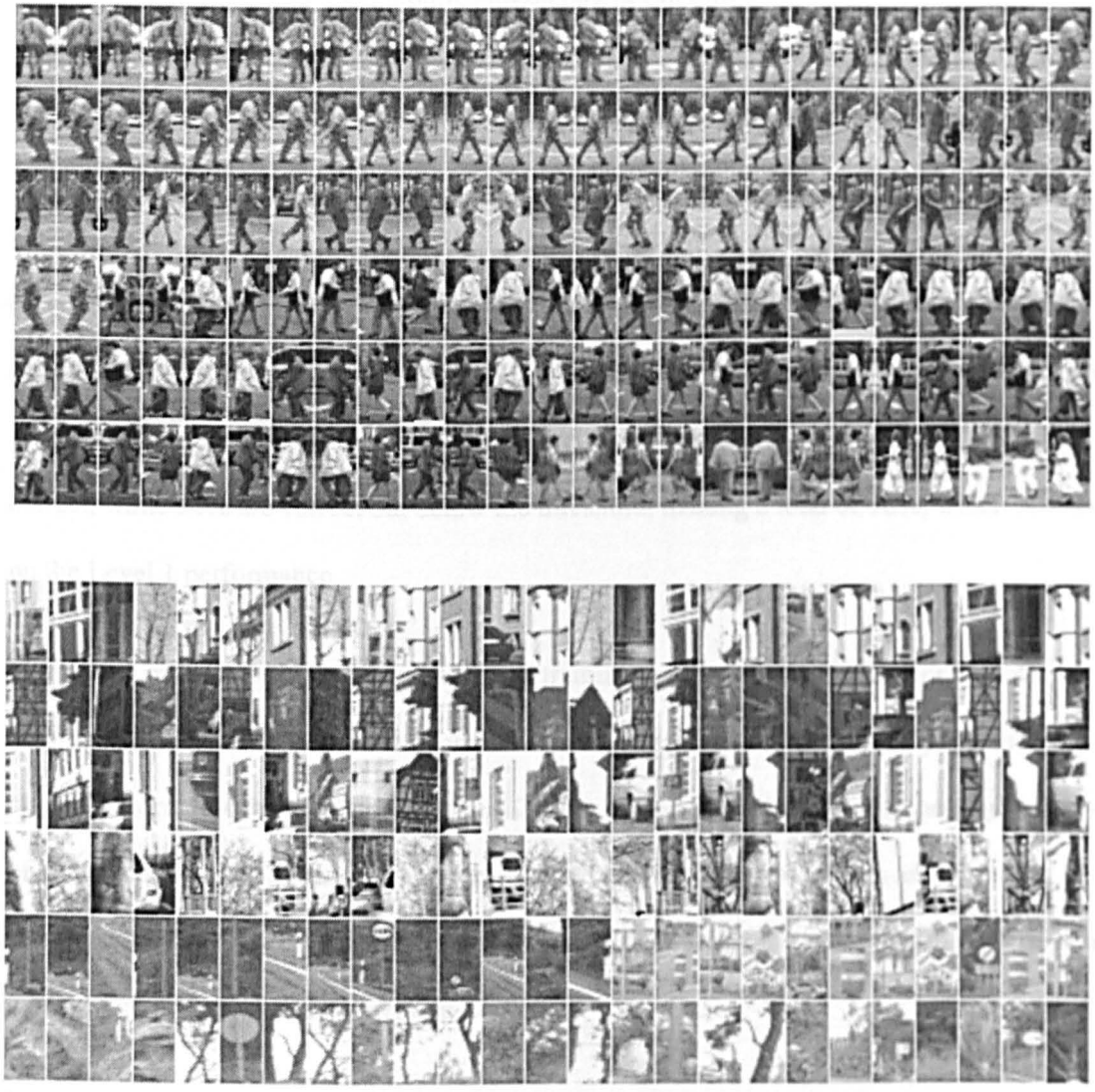


Figure 5.40: New test set consisting of 150 each of pedestrian and non-pedestrian images

In the first trial, the Level 2 training set was comprised of the full set of 100 training images, each represented by 80, 2-neighbour pairs, while for the second trial, the refined Level 2 data, selected with the help of the Incidence Matrix, was used. In both trials, just the 2-neighbours that disagreed on the classification are rematched and the overall score adjusted accordingly.

The Level 1 classification results are shown in the top section of Table 5.5. Without application of the 60% threshold, 83% + 12% = 95% of pedestrian images and 39% + 33% = 72% of non-

pedestrian images are correctly classified. It can also be seen that 83% of pedestrians and 39% of non-pedestrians score above the threshold and 1 pedestrian and 8 non-pedestrians are confidently misclassified. 24 pedestrians score less than 60% and so can be reclassified at Level 2, with just 6 of them being potentially correctable misclassifications. Among the non-pedestrians, 84 images score below the threshold, with 34 of them being potentially correctable.

The middle section of the table gives the Level 2 classification results when conflicting 2-neighbours are rematched against the full 100 training images, with each image represented by all eighty 2-neighbours. 89% of pedestrian images are confidently classified and 8% are correctly classified but below the threshold, giving a total of 97% correct classifications, which is an improvement on the 95% achieved at Level 1. 51% of non-pedestrians are confidently correct, while 25% are correct but below the threshold, making a total of 76%, which is 4% up on the Level 1 performance.

The results of using only the Matrix-selected training data for resolving the 2-neighbour ambiguity, appear in the lowest section of the table. Here we see that 91% of pedestrians are classified correctly above the theshold and 7% are correct but below the threshold, giving a total of 98%, which is 3% higher than at Level 1. Also, 59% of non-pedestrians are correctly classified above the threshold, while 21% are correct but below the threshold, making 80% in total, which is an 8% improvement on Level 1.

In these trials it can be seen that while both approaches to Level 2 classification improve on that at Level 1, refining Level 2 structure selection with the help of the Incidence Matrix can further enhance performance.

Level 1	Correct >= 60%	Correct < 60%	Non-class or incorrect < 60%	Incorrect >= 60%
Pedestrian	125 = 83%	18 = 12%	6 = 4%	1 = 1%
Non-pedestian	58 = 39%	50 = 33%	34 = 23%	8 = 5%

Using full training set with all 80 2-neighbours for each image

Level 2	Correct >= 60%	Correct < 60%	Non-class or incorrect < 60%	Incorrect >= 60%
Pedestrians	133 = 89%	12 = 8%	2 = 1%	3 = 2%
Non-pedestrian	76 = 51%	37 = 25%	21 = 13%	16 = 11%

Using Matrix-selected training data

Level 2	Correct >= 60%	Correct < 60%	Non-class or incorrect < 60%	Incorrect >= 60%
Pedestrian	136 = 91%	11 = 7%	2 = 1%	1 = 1%
Non-pedestrian	89 = 59%	31 = 21%	12 = 8%	18 = 12%

Table 5.5: Multilevel classification results for 50 pedestrian and 150 non-pedestrian test images

5.4.7 Summary of third set of experiments

The results of the trials in Section 5.4.6 indicate that, when a test image has been insufficiently confidently classified at Level 1, scoring less than the 60% threshold, its reclassification, at Level 2, is more likely to be correct, than that at Level 1. In addition, the results show that refining the Level 2 construct selection with the help of the Incidence Matrix can further enhance classification performance.

This has been a small-scale study, but these experiments suggest that an Incidence Matrix, implemented in a wrapper-based approach to feature selection, could potentially be helpful as a data-selection refinement tool, enabling a machine vision system to determine the 'usefulness' of particular types of structure at each level of representation/recognition in a given visual task, thus reducing combinatorial and dimensionality problems.

5.5 Fourth set of experiments: Autonomous construct generation for multilevel representation and recognition

In the fourth set of experiments the aim was to build a representation for hand-written numerals from the MNIST database, based on the heterogeneous polygons described in Chapter 4, Section 4.2.3, and further, to extend that representation to multiple levels, through the combination of lower level polygonal structures to form more complex constructs at the higher levels. The polygon-generation process was explained in Chapter 4, Section 4.2.3, and illustrated in Table 4.1, p129.

5.5.1 Applying the heterogeneous polygon-generating algorithm in binary images

The algorithm, described in Section 4.2.3, was applied to the 28x28 pixel numeral images. Figure 5.41(a) shows one of the polygons resulting from applying the algorithm to a '1'. Figure 5.41(b) shows some examples of the sets of polygons generated in response to the input of various numerals. The set of polygons returned when the algorithm is applied to a numeral is referred to here as the numeral's *polygon envelope*. Since the numerals are set against a plain background, the polygon envelope is able to provide some edge information. However, as pointed out in Section 4.2.3, the envelope is very variable, because the algorithm is sensitive even to small changes in greyscale, thus numerals from the same class can have very different response patterns as illustrated in Figure 5.42.

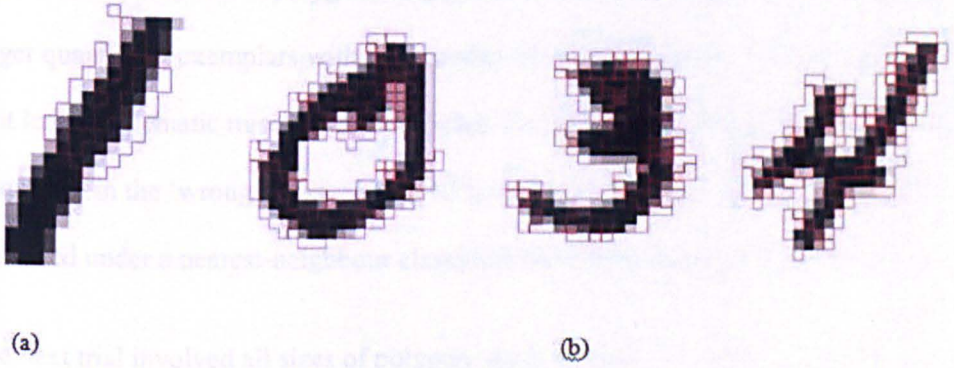


Figure 5.41: Applying the polygon generating algorithm
(a) One of the polygons generated by a '1'
(b) Polygon envelopes for different numerals

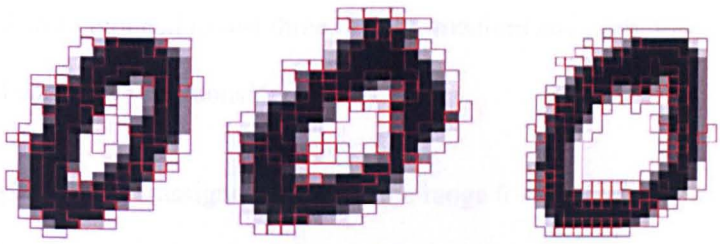


Figure 5.42: Within-class variability of the numeral envelope

5.5.2 First approach to making object descriptions more consistent within class

The polygons were encoded in terms of the number of 2x2s pixel patterns they contained, Figure 4.4, p135. To count the patterns in a polygon, a 2x2 window was 'slid' across it, being shifted by one pixel at a time, so that the sought patterns were overlapping, and whenever the window showed four pixels in a 2x2 configuration all belonging to the polygon, the 2x2 pattern count was incremented.

In the first trial, polygons containing just two 2x2 patterns, referred to here as size-2 polygons were considered and the numbers of 0s and 1s in the training set, 3000 examples of each class, that contained n size-2 polygons were determined – n ranging from 2 to 12, which are the most frequently-occurring quantities. Then 2000 test samples of 0s and 1s were categorized on the basis of how many size-2 polygons they contained, being assigned to the class that had the larger quantity of exemplars with that number of size-2 polygons. This was not very successful as it led to systematic misclassification when the particular number of polygons was found more frequently in the 'wrong' class. About 85% of the 1s but only 53% of the 0s were correctly classified under a nearest-neighbour classification scheme employed for all the experiments.

The next trial involved all sizes of polygon, again measured in 2x2s, but rather than considering each of the sizes from 2 to 58 2x2s - the minimum and maximum sizes across the training set, it

was decided to assign them to three categories – small, medium and large. To reduce the dimensionality of the problem still further, the sixteen 2x2s patterns, introduced in Chapter 4, Section 4.2.3 were reduced to just three – light, medium and dark, Figure 4.5, and initially only the light and dark sets were considered.

Now a polygon could be assigned a code in the range 0 – 5, according to whether it was small, medium or large and whether the number of light 2x2s was greater or less than the dark 2x2s. Each numeral would then be represented by a 7-dimensional vector, the first element of which was the number of polygons in the envelope and the other elements were the number of occurrences of the six different types of polygon. For example, a numeral with an envelope consisting of 9 polygons, of which two are of type 0, three are of type 1, two are of type 2, two are of type 3 and none are of type 4 or 5, would be stored as the vector <9, 2, 3, 2, 2, 0, 0>. In addition, it was decided that only images with envelopes containing the same number of polygons could be compared during classification, in a nearest neighbour classification approach, with 2000 test examples each of the 0s and 1s being compared with 3000 training examples of each class. Within this scheme, discrimination of 0s and 1s was slightly better over all than in the first experiment, with the 0s being about 83% and the 1s about 65% correctly classified, but performance was still low and poorly balanced between the classes.

So, for the next trial the representation included further relationships among the light, medium and dark 2x2s and the three polygon sizes. This time, a polygon would be categorized according to, for example, whether the number of light 2x2s exceeded the numbers of medium and dark 2x2s. This created seven categories of light/medium/dark relations and these, in conjunction with the three polygon sizes and the number of polygons in the image envelope, gave rise to vectors of 22 dimensions. This time about 80% of the 0s and 76% of the 1s were correctly recognized – still rather low scores albeit with a better balance between the classes. However, 0s and 1s should be relatively easy to discriminate to virtually 100% accuracy even just using simple pixel-matching, so these rather poor results strongly suggest that too much important information about the polygons was being discarded and that possibly the restriction to only matching polygon envelopes of the same size was too limiting. Furthermore, when the

1s were compared with the remaining numeral classes, the results were poorer still, ranging from about 30% to 60% discrimination accuracy.

5.5.3 Changing tactics

As noted in Chapter 4, Section 4.3.2, another major problem with the above approach is that no account was taken of the relative locations of polygons within the polygon envelope and so there was a danger that similarities found between numerals might be based on matching polygons from quite different regions. Therefore, instead of trying to learn a generalized envelope description for each numeral class, the system would derive a fixed set of 'polygon windows' through which to inspect incoming images.

For these trials it was assumed that only two classes of numeral had been introduced initially, and so the system was required to generate a set of polygons using just the 0s and 1s classes. Fifty examples of each class were used to generate the polygons and 1398 polygons containing at least four pixels were pooled and the modified version of the Relief Algorithm discussed in Chapter 4, Section 4.3.3, was applied to a randomly-chosen subset of 100 polygons from the pool using a random sample of 1000 items from the first 3000 exemplars from the 0s and 1s training sets.

The *Relief* algorithm was iterated 100 times, once for each polygon. On each iteration, a polygon was chosen for evaluation and reference image, R_i, was selected from the training items, drawing from each class on alternate iterations. The ten nearest-neighbours from the same class (hits) and from the opposing class (misses) were found, the distance being measured using the polygon currently being evaluated, as explained in Section 4.3.3 of Chapter 4. For each polygon window, ten cycles of choosing a reference image and nearest neighbours were performed and on each cycle, the polygon's score adjusted according to how close the reference is to its nearest neighbours. If, on a given cycle, the hits were nearer than the misses, the polygon's 'usefulness' score was increased. Finally each polygon score was totalled and an

average found and then the forty-four polygons with an above average score were stored for further evaluation.

The next step was to test the classification ability of each of the forty-four 'above-average' polygons, using the first 3000 training images for each of the 0s and 1s classes for classifying the next 2000 images from each training set. Table C.1, Appendix C, shows the results.

The table shows each polygon's number in the original pool and gives its size well as its classification scores. The better performers are highlighted in bold, polygons with a double asterisk scoring highly on both classes and those with a single asterisk performing highly on one class, but only moderately well on the other. The results show that there are nine high-scoring polygons, with polygon 1166 being a nearly perfect classifier for the data.

This highest scoring polygon was then tested on the ten numeral classes, again using a nearest neighbour approach with 3000 training examples and 2000 test examples of each class. Examples were compared through pixel-matching. The results can be seen in Table 5.6.

0s	1s	2s	3s	4s	5s	6s	7s	8s	9s
88.4	96.1	69.3	80.0	86.5	84.2	90.8	84.0	63.9	65.7

Table 5.6: Percentage scores for all ten numeral classes classified by the 'best' polygon, in an absolutely fixed position within the 28x28 image frame

Performance is variable over the ten classes, but is quite good for a single polygon classifying 20,000 examples. However, the disadvantage of this approach is that it is not translation invariant. In order to achieve translation invariance, the polygons would need to be positioned relative to the 'centre' of the object, which in this work was taken to be the centre of the box enclosing the numeral envelope. Thus a polygon was positioned in every image in the same location relative to the envelope centre as it had occurred in the originating image. Figure 5.43 shows a sample of numerals with the 'best' polygon positioned relative to the envelope centre.

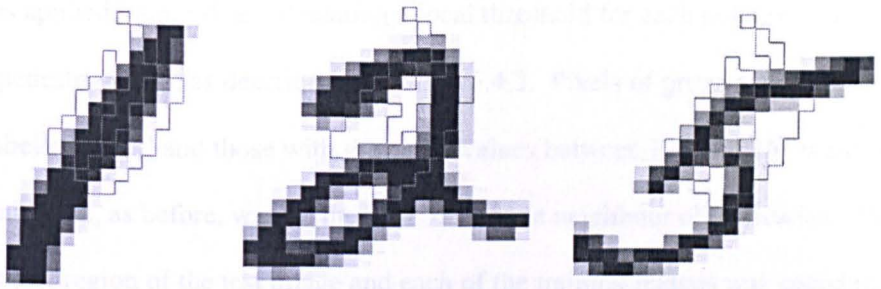


Figure 5.43: Examples of numerals with the 'best' polygon positioned relative to the centre of the polygon envelope. The '1' is the numeral which originally gave rise to the polygon.

This relative positioning of the polygon brought about some reduction in classification accuracy since the envelope-box centres are sensitive to small changes in the envelope, which in turn affects the positioning of the polygon. Table 5.7 shows the results for the same polygon with the ten classes.

0s	1s	2s	3s	4s	5s	6s	7s	8s	9s
78.6	90.3	54.9	68.8	75.1	78.2	79.9	47.6	59.1	16.5

Table 5.7: Percentage scores for all ten numeral classes classified by the 'best' polygon, located relative to the centre of the numeral envelope box rather than in an absolutely fixed position, expressed in percentages

Although performance was considerably reduced, it was a necessary sacrifice in order to gain translation invariance and the ability to locate and examine multiple objects within an input scene. Also, it would be expected that the system would have to employ several polygons to extract sufficient information about these shapes to be able to classify them reliably.

5.5.3.1 Growing a representation

The next trials set out to determine whether the addition of more polygons, as the system was introduced to more and more objects would in fact aid discrimination. It was decided to encode polygons as 16-dimensional vectors of 2x2s pattern counts from this point on, to provide some information on the structure within the polygon windows and allow greater generalization. To

facilitate this, because the images are effectively binary, a simple global greyscale threshold of 128 was applied, rather than calculating a local threshold for each polygon window, as in the NiSIS pedestrian data, as described in Section 5.4.2. Pixels of greyscale value less that 128 were labelled 'dark' and those with greyscale values between 128 and 255 were labelled 'light'. Discrimination, as before, was on the basis of nearest neighbour classification. For a given polygon, its region of the test image and each of the training images was encoded as a16-dimensional vector of 2x2s pattern-counts. The test vector was compared with all the training vectors in turn, and the closest match selected. These closest matches were then summed for each class, across the current set of discriminative polygons and the test image assigned to the class with the smallest total.

The process began with just two classes and a single polygon with which to learn to classify them. The idea was that if classification was sufficiently accurate above a certain threshold, a new class would be introduced into the repertoire, and the system would attempt to classify all three classes using the same single polygon. If performance deteriorated by a significant amount, a new polygon would be selected from the set of forty-four 'successful' polygons and used in conjunction with the first in a second attempt to classify the three classes. Then if performance improved sufficiently, another new class would be introduced and the same polygons used to classify all the classes currently in the repertoire. If performance did not improve enough with the new polygon, another new polygon would be chosen to replace it and another attempt at classification would be made. This process would continue until the classification repertoire contained all ten numeral classes. Table 5.8 shows how classification ability fluctuated as the number of classes and polygons steadily increased. The final set of polygons had eight members, all of them having originated from the image envelopes of the originally selected subset of 0s and 1s. Most were chosen from among the forty-four 'successful' polygons, but the final two were selected at 'random' from the original pool.

Polygons				Classes	%	correct				
	0s	1s	2s	3s	4s	5s	6s	7s	8s	9s
1	84.7	86.9	-	-	-	-	-	-	-	-
	54.5	54.9	45.0	-	-	-	-	-	-	-
2	78.2	65.1	70.5	-	-	-	-	-	-	-
	74.5	64.4	62.0	66.4	-	-	-	-	-	-
3	79.4	71.9	69.7	71.35	-	-	-	-	-	-
	78.5	63.8	60.0	68.4	61.4	-	-	-	-	-
4	82.4	63.6	67.4	75.8	64.1	-	-	-	-	-
	81.0	62.2	65.9	61.7	62.3	56.4	-	-	-	-
	77.2	61.0	63.0	57.7	60.9	50.0	66.9	-	-	-
5	79.4	73.7	67.6	62.4	66.7	54.6	73.0	-	-	-
	79.2	69.7	66.4	61.7	57.6	54.2	73.0	52.7	-	-
6	80.6	77.2	70.3	70.9	65.3	67.9	78.5	58.2	-	-
	80.3	74.8	68.2	65.5	61.3	63.8	73.5	58.1	64.8	-
7	80.8	81.7	72.2	67.8	71.1	60.8	72.4	66.1	68.8	-
	80.7	81.3	72.0	67.2	56.9	60.1	72.4	59.3	63.3	46.6
8	82.1	84.9	74.4	71.4	60.7	63.0	74.9	67.3	71.0	53.8

Table 5.8: Classification accuracy as the number of classes and polygons is increased

Table 5.8 confirms the expectation that introduction of a new class reduces performance for a given set of polygons, and addition of a new polygon tends to enhance performance for a given set of classes.

The graph in Figure 5.44 below shows the overall effect on the system's classification performance of the gradual introduction of new classes and polygons at the first level of processing. It illustrates the fluctuation in average classifications across the 'known' classes as the number of classes and polygons increases. The final two data points, labelled L2 and L3 on the abscissa, show the effect of combining polygons to form higher-level structure, as will be discussed next.

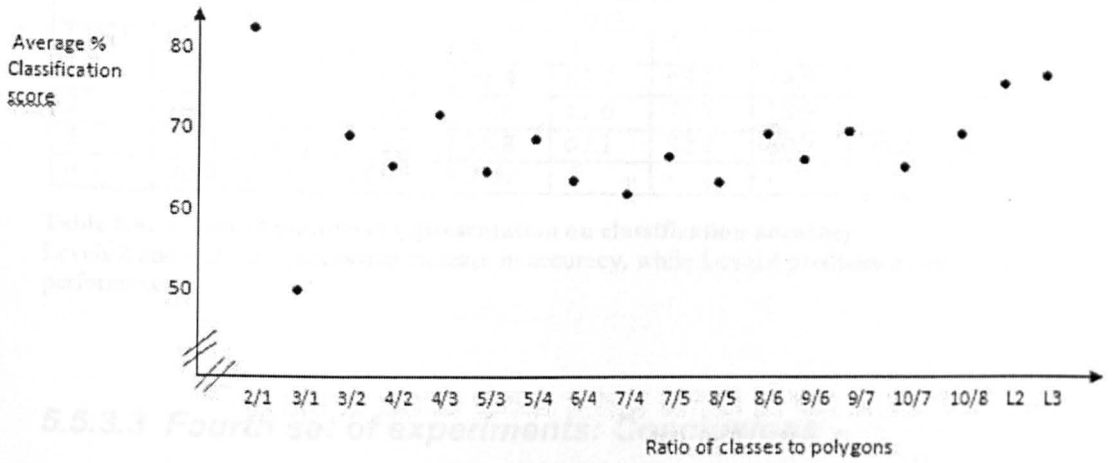


Figure 5.44: Classifying hand-written numerals at multiple levels
Average classification scores as numbers of classes and polygons increase at representation/recognition Level 1. L2 indicates Level 2 at which there are four constructs which are pairs of polygons and L3 indicates Level 3 where there are two constructs which are pairs of the Level 2 pairs of polygons and the classification result at these higher levels is the average over all ten classes.

5.5.3.2 Multilevel representation

In order to investigate whether higher-level representation could improve classification performance, the next set of trials added, step-by-step, a further three levels of representation above that of the individual polygons at Level 1. At the first level, each polygon had contributed to the classification output by assigning the part of the input test numeral it could 'see' to the class of the training image providing the closest match. At Level 2, the eight polygons were combined on the basis of closest neighbours, into four pairs. Classification was now dependent on both members of a pair contributing to the closest match within the one training image. The constructs at Level 3 were two subsets of four polygons – pairs of the pairs from Level 2. These third-level pairs were now required to find a closest match with a single image. Finally, at Level 4, the single construct, comprised of all eight polygons, must now be matched to the one image. Table 5.9 contains the Levels 1, 2 and 3 classification scores for all ten classes. Overall performance is shown to improve slightly at successive levels up to Level 3. The results for Level 4 are only partially represented, as it was apparent that accuracy was being reduced at this level. The data points 'L2' and 'L3' in Figure 5.44 show an overall improvement at successive levels across all classes, in particular at Level 2.

					Class					
Level	0	1	2	3	4	5	6	7	8	9
1	82.1	84.9	74.4	71.4	60.7	63.0	74.9	67.3	71.0	53.8
2	90.0	94.0	80.4	75.0	67.6	73.9	78.9	77.5	73.8	62.4
3	90.8	95.5	78.0	75.8	67.1	82.2	80.2	78.9	75.2	64.9
4	90.0	95.3	80.5	74.9	-	-	-	-	-	-

Table 5.9: Effect of multilevel representation on classification accuracy
Levels 2 and 3 show a successive increase in accuracy, while Level 4 produces a slight reduction in performance.

5.5.3.3 Fourth set of experiments: Conclusions

This work has demonstrated that polygonal structures generated by applying a simple region-growing algorithm to objects on a plain background have the potential to provide useful information for object discrimination, thus enabling an object recognition system to derive meaningful representations with minimal reliance on user input.

It was found that the polygon envelope that the algorithm produces can vary considerably among objects of the same class as well as across different classes. Hence the main part of the work has been focussed on restricting the representation to a small but expandable subset of 'fixed' polygons that could be applied to multiple classes of object, making object descriptions more stable and simplifying the derivation of higher-level structure.

The experimental results show that, although classification rates are not particularly high, the polygons are nevertheless enabling the system to classify well above chance and that classification rates can be improved to a degree by adding more polygons and combining them to form higher levels of representation.

The pool-generation and polygon-selection process was done on a very small scale, only making use of polygons extracted from the 0s and 1s classes. Performance could well be enhanced by forming the initial pool of polygons from a larger number of classes.

A major question, addressed in the next set of experiments, is whether this approach of extracting heterogeneous polygons can be adapted to other computer vision applications especially those involving cluttered scenes.

5.6 Fifth set of experiments: Building a multilevel heterogeneous polygon representation in cluttered scenes

In the fifth set of experiments the heterogeneous polygon generating algorithm was applied to the NiSIS pedestrian dataset with the aim that, as in the experiments with MNIST data, the system would abstract its own constructs with which to represent the objects it was learning to discriminate, in this case, images containing a pedestrian from those that do not.

One obvious difference between this task and numeral recognition is that there are only two classes to consider instead of ten. In this respect, the problem seems easier, however, pedestrians form a large class, the members of which can vary considerably in appearance due to differences in clothing, pose, orientation, size, shape, lighting and so on, while non-pedestrian images constitute a potentially infinite variety of scenes, so as well as the considerable variability, there is also an imbalance in size between the two classes.

An additional problem for the algorithm is that it relies on an uncluttered background to be able to segment out the object of interest. However, given that the pedestrian images are not set against a plain background and the non-pedestrian images consist mainly of cluttered scenes, the algorithm would indiscriminately grow polygonal regions right across the images and polygons would be likely to frequently span both foreground and background in the images containing pedestrians. This raised the question of whether the system could find any constructs that would make good discriminators from among the many polygons generated in response to this type of image and further, whether, if a set of 'good discriminator' polygons could be found, would higher-level structure formed by combining polygons into more complex constructs enhance the system's performance.

5.6.1 Applying the algorithm to the Daimler-Chrysler database

As described in Section 4.2.2, the images in the DaimlerChrysler dataset are low-resolution greyscale images of dimension 36 pixels high by 18 pixels wide, showing either a pedestrian against a busy 'street' background or a 'cluttered' non-pedestrian scene. Figure 5.45 provides a small sample of pedestrian and non-pedestrian images used in these experiments, and Figure 5.46 shows the effect of applying the algorithm.

The database contains a substantial number of images that are to a greater or lesser extent 'greyed out' presumably to introduce the concept of occlusion to the competition. However, this work was not concerned with the problem of occlusion and so these images were excluded from the training and test sets used in the experiments. This left 304 pedestrian and 547 non-pedestrian images in the training set and 582 pedestrian and 1102 non-pedestrian images in the first test set. Later, images from the much larger second competition test set were used to augment the initial training and test sets, as well as to provide further unseen data for a final test of the system once the process of polygon selection and multilevel representation was complete.

Figure 5.45: Sample of pedestrian and non-pedestrian images from the training set

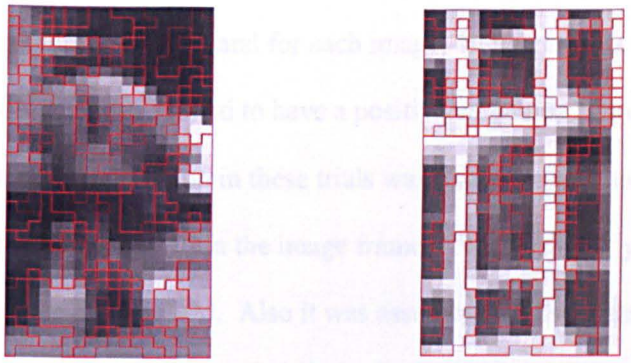


Figure 5.46: Effect of the region-growing algorithm to a pedestrian and a non-pedestrian image also shown in Figure 4.3

As anticipated, the polygons cover the entire image, and in the pedestrian image, some of them span extended regions of background in conjunction with foreground especially in the leg area.

5.6.2 Polygon generation and selection

A pool of 27,172 polygons was generated by applying the algorithm to the 304 pedestrian and 547 non-pedestrian training images. Of these, 8834 were found to be above the average size of 13 pixels and so these were retained to be encoded as 16-dimensional vectors of 2x2s pattern counts. Since the images are not binary, it was necessary to 'binarize' each polygon. Within a particular training image, the average greyscale value of the pixels comprising a given polygon was used as a threshold, so that pixels darker than the average would be set to '0' or black and those lighter than the average would be set to '255' or white. A 16-D vector of 2x2s pattern counts was derived for every polygon's instantiation in every training image. Any polygons for which fewer than ten 2x2s patterns could be extracted were rejected at this stage. This reduced the number of eligible polygons to 201, thus providing a total of 304x201 = 61104 vectors representing the polygons' instantiation in pedestrian images and 547x201 = 109947 vectors representing their 'non-pedestrian' instantiations.

The modified version of the *Relief* algorithm, as described in Section 5.4.3, was then applied to the task of determining which polygons might make good discriminators of the two classes. 201 polygons that were considered large enough to potentially provide useful image

information, being comprised of at least 10 2x2s patterns, were evaluated. For each polygon, 20 images, R_i, were randomly selected and for each image, 10 nearest hits and 10 nearest misses calculated. 151 polygons were found to have a positive weighting score and were selected for further evaluation. The assumption in these trials was that there was no significant variation in the location of the pedestrians within the image frame. That is, the (x, y) locations of the selected polygons were totally fixed. Also it was assumed that the scale of the pedestrians did not vary to any extent. Figure 5.47 indicates the location of two of the best-scoring polygons within the 36x18 image frame.

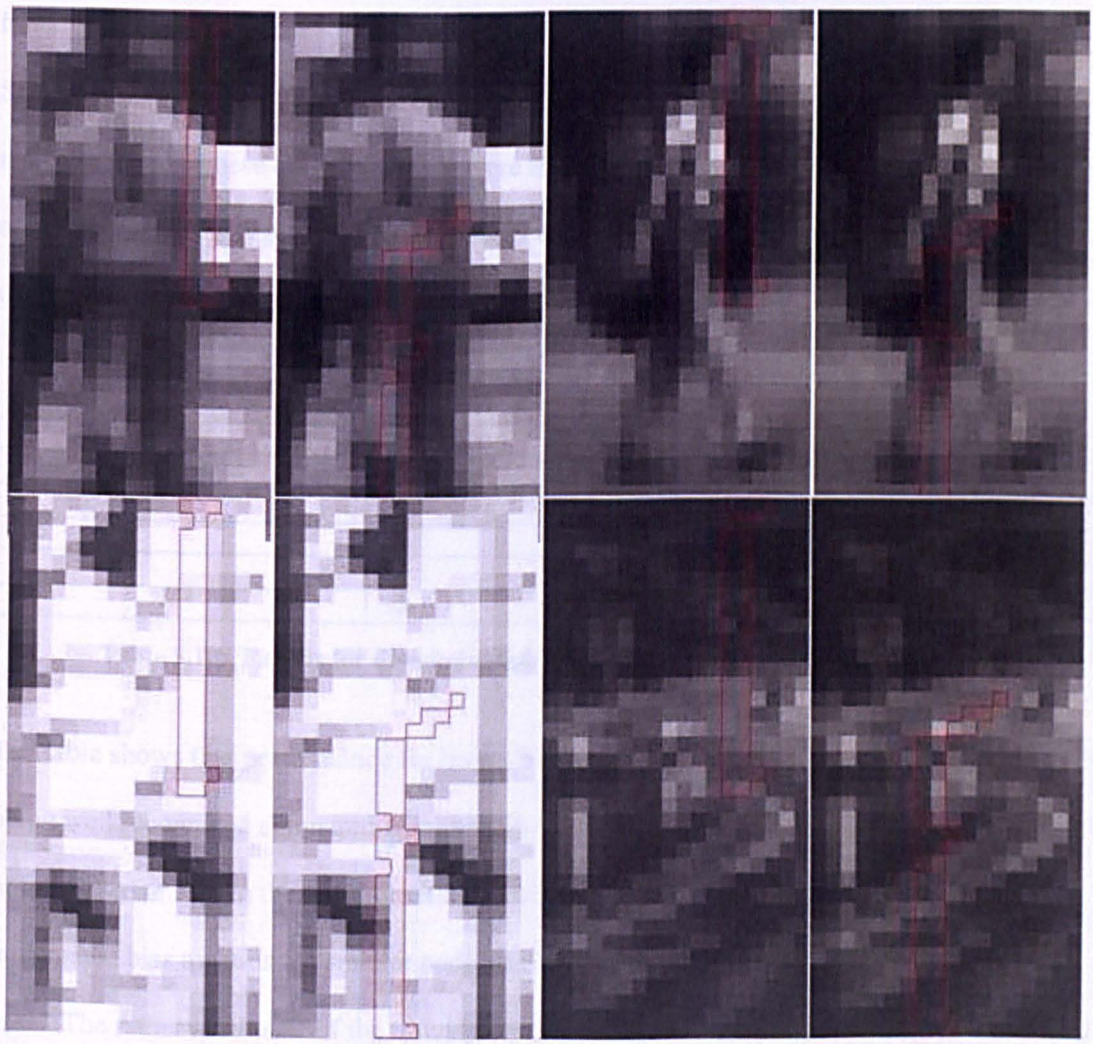


Figure 5.47: Two of the best discriminating polygons
shown on two pedestrian and two non-pedestrian training images. As in Figure 5.46, it can be seen that in the pedestrian images the polygons enclose a considerable area of background

5.6.3 Building the set of discriminative polygons

The process began with the testing of the highest-scoring polygon output from the Relief Algorithm using the 582 pedestrian and 1102 non-pedestrian images in the first test set, and then, one by one, more polygons were introduced in descending order of their 'Relief' score and tested in the same way. The idea was that, if a polygon enhanced discrimination it would be retained, otherwise it would be rejected. Discrimination was again on the basis of nearest neighbour classification. For a given polygon, its region of the test image and each of the training images was encoded as a16-dimensional vector of 2x2s pattern-counts, as described in Section 5.5.2. The test vector was compared with all the training vectors in turn, and the closest match selected. These closest matches were then summed for each class, across the current set of discriminative polygons and the test image assigned to the class with the smaller total. The results for the first four polygons are presented in Table 5.10.

	% scores	
Number of polygons	Pedestrians	Non-pedestrians
1	28.0	48.0
2	33.5	61.6
3	35.1	66.7
4	38.1	72.0

Table 5.10: Results for discrimination sets comprised of 1, 2, 3 and 4 polygons

The table shows that performance for both classes was increasing, but the pedestrians were not being well recognized compared with the non-pedestrians. It had been thought that having more non-pedestrian data in the training set than pedestrian data might have helped redress an anticipated bias towards the smaller pedestrian class. However, this did not appear to be the case. The rather small size of the training sets could be contributing to the problem, therefore it was decided to 'boost' the pedestrian training data by augmenting the set with test examples the system had failed to discriminate. 343 such examples were transferred from the test set to the training set, and to replenish the pedestrian test set and augment it to the size of the non-pedestrian test set, 763 examples were taken from the second test set mentioned in Section

5.6.1. Table 5.11 shows the results for four polygons, increasing to ten, with the adapted pedestrian datasets.

	% score	
Number of polygons	Pedestrians	Non-pedestrians
4	67.8	49.0
5	71.2	53.8
6	71.3	59.1
7	73.1	59.6
8	75.2	62.1
9	75.7	64.7
10	76.3	64.3
10	76.7	66.3

Table 5.11: Gradually increasing the number of polygons in the discrimination set from 4 to 10

The entries in the table show firstly that the 'balance' between the classes in terms of performance shifted in favour of the 'boosted' pedestrians. However, the discrepancy was not so great as before, and it narrowed as both classes improved steadily with the gradual introduction of more polygons, until the 10th polygon was added. At this point, due to the slight decrease in non-pedestrian performance, the 10th polygon was substituted and the replacement enhanced the performance for both classes above that for 9 polygons.

In Table 5.12 the result of gradually increasing the number of polygons from 10 to 16 is shown.

	% score		
Number of polygons	Pedestrians	Non-pedestrians	Average of both classes
10	76.7	66.3	71.5
11	77.0	65.3	71.2
11	77.1	66.0	71.6
11	75.8	68.2	72.0
12	76.4	68.4	72.4
13	78.0	70.2	74.1
14	78.3	71.8	75.1
15	78.4	71.1	74.8
15	78.2	72.7	75.5
16	80.4	72.3	76.4

Table 5.12: Gradually increasing the number of polygons in the discrimination set from 10 to 16

It can be seen from Table 5.12 that it became more difficult to improve performance in both classes as new polygons were added. One possible reason for this is that the further down the list of outputs from the Relief Algorithm the polygon came, the less reliable it was likely to be.

Another reason might be that it was getting harder for each successive polygon to provide significantly new information.

The first entry in the table is a repeat of the final entry in Table 5.11 for ten polygons, to show how the first choice of the eleventh polygon reduced performance in the non-pedestrians, while slightly improving the pedestrians. The third choice of eleventh polygon was selected because, although the pedestrian score was reduced, that of the non-pedestrians improved and the average of the two classes was the best of the three. Both classes then enjoyed a slow but steady improvement in performance until the fifteenth polygon was chosen. The second choice was selected because, although the pedestrian score was slightly down on that with the first choice, the non-pedestrians improved and the average score was increased. Finally, with the sixteenth polygon the non-pedestrians were slightly down but the pedestrians were up as was the average of the two.

5.6.4 Multilevel representation

Beyond the sixteen polygons already selected, it looked unlikely that performance could be improved by adding further individual polygons, therefore it was decided that this would be a good point at which to attempt to answer the second question, as to whether higher level structure, composed at each successive level of combinations of the previous level's constructs, could be used to improve discrimination. Level 2 representation would consist of eight pairs of polygons. A polygon would be paired with its closest neighbour in terms of the smallest distance between the centre of its bounding box and those of its neighbours. At Level 3, four constructs would be formed by pairing off the pairs of polygons from Level 2 with their closest neighbours. Two Level 4 constructs would consist of pairings of the Level 3 structures and at Level 5, there would be a single construct. At each successive level, it would be required that the associated constructs would have to be matched within a single training image. Table 5.13 shows the results for the new compound constructs at Levels 2 and 3.

	% score		
Level	Pedestrians	Non-pedestrians	Average of both classes
2	87.5	69.8	78.7
3	94.0	68.2	81.1

Table 5.13: Results for the new compound constructs at Level 2 and Level 3

At both Level 2 and 3, pedestrian performance is significantly increased over that at Level 1, but unfortunately this is at the expense of the non-pedestrians. Thus it was decided to 'boost' the non-pedestrian training set with 100 test examples the system had misclassified, and to replenish the test set with 100 non-pedestrian images from the second test set. Table 5.14 shows the impact of the 'boosting' at Level 3 and the results with the modified data set at Levels 4 and 5. At Level 4, the three possible pairings of the Level 2 constructs were tried.

	% score		
Level	Pedestrians	Non-pedestrians	Average of both classes
3 (before boosting)	94.0	68.2	81.1
3 (after boosting)	92.0	80.6	86.3
4	91.8	72.5	82.2
4	95.2	75.0	85.1
4	93.8	71.1	82.5
5	95.3	70.0	82.7

Table 5.14: Impact of non-pedestrian boosting on performance at Level 3 and results with the modified data at Levels 4 and 5

While pedestrian performance was reduced a little, the boosted non-pedestrians' score was a significant improvement on the Level 2 and Level 3 scores before boosting. For the second and third pairings at Level 4, the pedestrian score exceeded that at Level3, but non-pedestrian performance was well down and the average scores for the two classes were lower than that at Level 3. Level 5 produces the highest score of all the levels for the pedestrians, but again, at the expense of the non-pedestrians. It may well be that further boosting of the non-pedestrian training set could help overcome the continuing imbalance between the classes.

Although new test data had been introduced to both classes at different stages during the experiments, all the current test data had been used to verify that the system had improved its discriminatory ability as new polygons were tried or a new processing level was added.

Therefore it was necessary to test the system with some completely unseen data. There were now 708 pedestrian images, that did not contain any occlusion, left in the second test set, so it was decided to form a new test set comprised of these and 708 of the remaining non-pedestrian images from that set. The system's performance with these previously unseen images at Levels 3 to 5 is shown in Table 5.15.

	% score		
Level	Pedestrians	Non-pedestrians	Average of the 2 classes
3	93.1	77.1	85.1
4	95.6	71.3	83.5
5	96.5	67.0	81.7

Table 5.15: Classification results for the new test set at Levels 3 – 5

Figures 5.48 and 5.49 provide a graphic representation of performance at all levels. Figure 5.48 summarizes the results in Tables 5.11 – 5.12 for steadily increasing the number of polygons at Level 1.

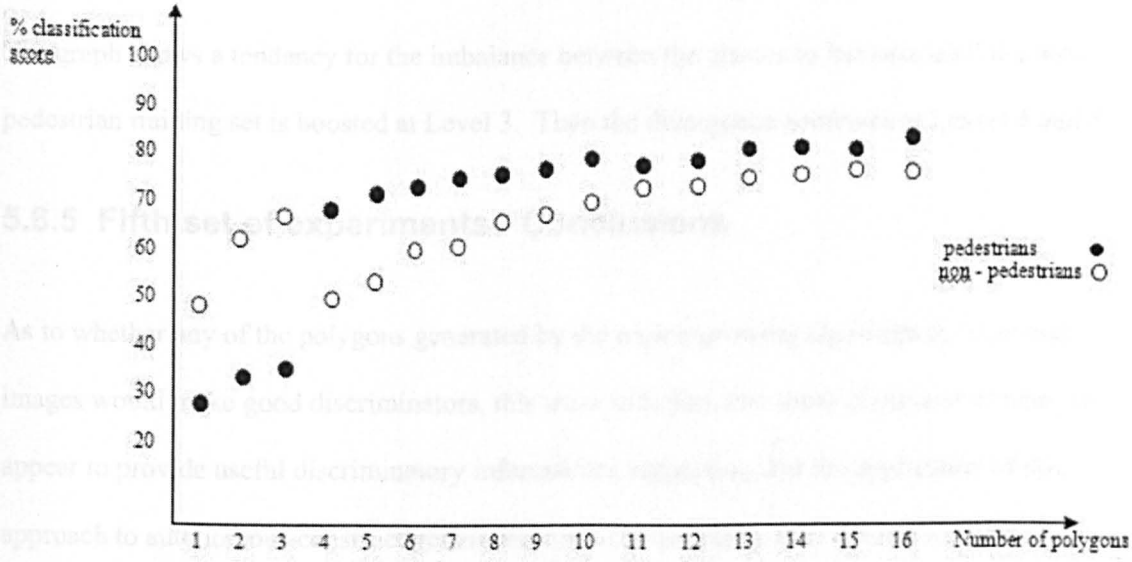


Figure 5.48: Overall increase in discriminatory performance
with pedestrian and non-pedestrian images as more polygons are added. The 'crossing-over' of data points at the 4th polygon on the graph shows the effect of boosting the pedestrian training set. Then, as the polygon set grows, the effect of the imbalance in performance between the classes tends to be reduced.

Figure 5.49 illustrates the results for classification at five different levels, tabulated in Tables 5.13 – 5.15.

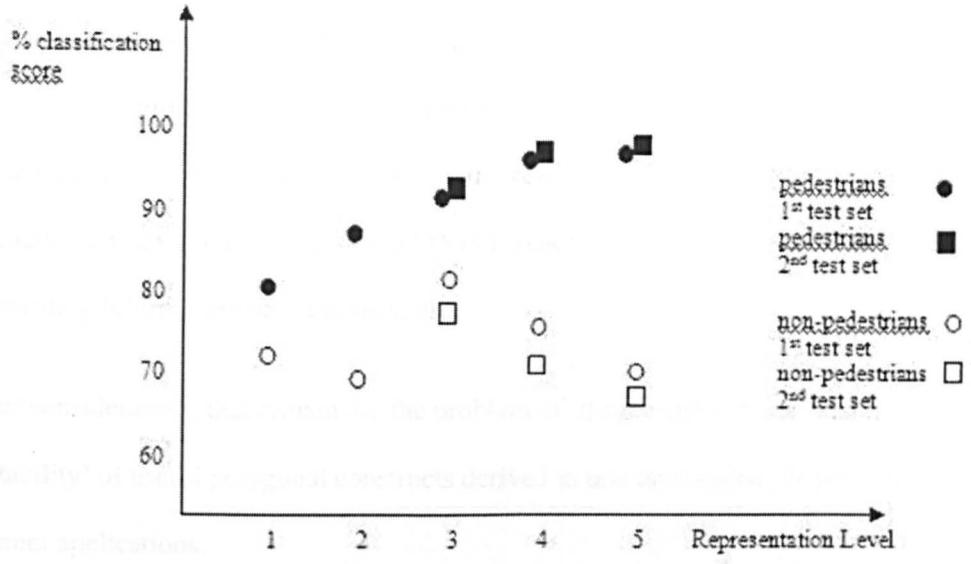


Figure 5.49: Classifying the pedestrian and non-pedestrian images at multiple levels

The pattern is similar for both test sets. Pedestrians continue to improve at each successive level. Non-pedestrians performance drops at Level 2, but is considerably improved after boosting at Level 3. However, at Levels 4 and 5 it deteriorates again.

The graph shows a tendency for the imbalance between the classes to increase until the non-pedestrian training set is boosted at Level 3. Then the divergence continues at Levels 4 and 5.

5.6.5 Fifth set of experiments: Conclusions

As to whether any of the polygons generated by the region-growing algorithm in 'cluttered' images would make good discriminators, this work indicates that some of these structures do appear to provide useful discriminatory information, suggesting that the application of this approach to autonomous construct generation need not be restricted to objects on plain backgrounds.

The experimental results also suggest that higher level constructs, formed by combining polygons that perform well individually into more complex structures, can, up to a limit, further enhance discrimination, although there is a tendency for imbalances in the data to become more pronounced at higher levels.

Some of the factors that would be likely to influence the success of a polygon are its location, size and shape, and the degree of spatial overlap with its neighbours. In this work, size and shape, in terms of the number of 2x2s patterns a polygon contained were considered, but no account was taken of location or overlap in the selection of the discriminatory set. In addition, the encoding in 2x2s patterns, is in the MNIST experiments, would have provided only fairly rudimentary information on local structure.

Major considerations that remain are the problem of image segmentation and also the issue of 'portability' of useful polygonal constructs derived in one application, or perhaps several, to different applications.

Chapter 6: Conclusions

The motivation for the thesis has been the challenge of developing machine vision systems that are fully autonomous, with minimum 'engineering' intervention from the system designer. Fundamental requirements for autonomy in such systems are being able to learn and adapt as environmental demands change.

In this chapter, conclusions are drawn from the experimental results of Chapter 5, with respect to the issues raised in Chapter 1 and the research questions formulated at the end of Chapter 3. The contributions of the thesis are indicated and, in addition, possible approaches to tackling some further questions that have arisen as a result of this work are suggested. Finally a recent development in the automatic formation of multilevel representations, Johnson, (In Press), is described and some questions arising from the associated early work are discussed.

6.1 Answering the research questions

In addressing the research questions, certain key areas in which many artificial object recognition systems tend to rely on engineered approaches were identified in Chapter 4. These areas are ***feature extraction***, ***feature selection*** and the ***representation architecture.***

The approaches to the research in these areas and its outcomes are now reviewed in the light of the research questions.

Question 1: Is there a general architecture for representing multilevel systems, the same 'formula' being appropriate for a wide variety of representation/recognition problems?

The work of the thesis suggests that a hypernetwork-based representation can provide such a general architecture.

In the second set of experiments, this hypothesis was tested by building a multilevel representation for simple geometric shapes. The visual task, to discriminate two classes of object, was not demanding, but was sufficient to demonstrate how multiple levels can be

constructed by repeated application of the fundamental principle of hypernetworks, Chapter 4, Section 4.5.1.

In addition, the representation of the structure at each level can be made explicit, so that information about the nature of the representation at any level is readily accessible and can be used for classification, as demonstrated in the second, third, fourth and fifth sets of experiments. In the second set of experiments, classification was attempted first at the whole object level, and then at the level of the individual 'curvature' constructs. Subsequently, classification at intermediate levels was effected by allowing the classification of an individual construct to be influenced by that of its left and right neighbours. Classification results showed that being able to access intermediate-level structure in this general architecture improved performance.

In the third, fourth and fifth sets, this same architectural framework was applied successfully in the classification of hand-written numerals and pedestrian recognition tasks. In the fourth and fifth sets, classification performance was not high, in part, due to the nature of the heterogeneous polygons and their descriptors, however, it was demonstrated that even through a fairly crude process of successively pairing constructs at successive levels under a spatial constraint, performance was increased at each higher level up to a limit, beyond which the classification was dependent on what was becoming increasingly like whole-object matching.

Thus this crude approach does find higher-level structure that 'exists' in the training images, however, it does not guarantee that it will be maximally useful for the task, in the sense of being 'class-specific'. Finding 'useful' higher-level structure was discussed in Section 4.5.3 and is revisited in the analysis of the work relating to Question 4 below.

In biological vision, it is thought that the representation becomes increasingly general further up the hierarchy. As discussed in Chapter 2, Section 2.3, neurons with large receptive fields such as those in IT are generally less sensitive to various transformations such as scale and rotation. In this work, limited generalization was achieved through the requirement of matching more complex constructs representing larger regions of a test image within a single training image, Section 4.5.6.

Question 2: Can such systems be self-forming?

The work of the thesis suggests that this is possible in a hypernetwork framework.

Two aspects of this question were explored in the thesis, firstly the way systems can create a representation in response to input data, and secondly, how they can adapt that representation as task requirements change by forming new constructs at new levels.

In the process of applying the fundamental principle of hypernetworks, a hypernetwork system forms simplices, the vertices of which represent the entities for which a particular relation or set of relations holds. Thus a simplex is a construct that binds the entities at a given level into structure at the next level, as explained in Chapter 4, Section 4.4.1.

These simplices are interconnected in a lattice formation, Section 4.5.1.1, reflecting shared structure. There are two complementary ways in which a system can form a representation of the input data, as a simplicial complex in which each object is depicted by a simplex, the vertices of which are the features or constructs that comprise it, or as the conjugate complex, in which each feature or construct is depicted by a simplex, the vertices of which are the objects in which that feature appears, Section 4.5.3.

So, in response to the input images, once a representation at the level of a suitable individual construct has been abstracted, the system forms a set of simplices each representing an object, and a set of simplices, each representing a construct.

The second set of experiments explored the construction of such a system. Analysis of the representation of the 'curvature' constructs as hubs with associated 'star' simplices of their associated training objects, showed the interconnected structure that could be formed automatically, on the application of a specific 'connectivity rule', Section 5.3.1.3, to these constructs in the process of their abstraction from the training data. Hubs then became apparent at multiple levels of representation, including that of the individual curvature constructs and of various combinations of them, as illustrated in the Incidence Matrix, Table 5.1. This was

possible because the formation of the star simplices of the hub constructs required that the constituent objects match exactly in terms of the hub.

In the third set of experiments, the vertices of an object simplex each represented one of the object's 208 window instantiations, while in the conjugate simplicial complex, the vertices of a window simplex represented all the training objects for which the window's description was the same length. In this case, other than the requirement that the *length* of description of the objects in a hub window's star be matched, there was no necessity for any of the descriptor values to match.

The stars of hubs representing pairs of windows were formed through the requirement that each constituent training object was instantiated jointly by the pair as the closest match to a particular test object. Thus the star could contain objects with different lengths of description, not all of which would be associated with correct classification, Section 5.4.5.2. This process was termed 'inexact construct matching', Section 4.5.4.

In this work, systems were able to adapt their architecture by ignoring irrelevant constructs (as in the second set of experiments), adding new constructs (as in the fourth and fifth sets of experiments), or forming new levels of representation (as in the third, fourth and fifth sets). This behaviour was prompted by inadequate classification performances in the fourth and fifth sets of experiments, and the need to resolve a classification conflict, in the second and third sets, Section 4.5.7. Self adaptation can also be prompted by the system user, as explained in Section 6.3.2.

Question 3: How can systems find their own descriptors?

The thesis made an initial attempt at enabling a system to find 'non-engincered' features, with limited success. Two approaches to autonomous construct abstraction were tried, the first to generate features randomly and the second, to apply an image-segmentation algorithm.

The random approach of the first set of experiments attempted to simulate autonomy through the generation of minimally engineered, highly-generic pixel-pair features. The results show that, with a small set of 'toy' shapes, such features can classify quite well.

It has been seen in Chapter 3 that many machine vision systems randomly generate features of different shapes and sizes and then select a subset of the more reliable ones. What is different in this work is that the pixel-pair features were able to vary randomly in terms of which of the four possible light-dark patterns they represented and the spatial relations between their constituent pixels.

However, these particular features would seem unsuitable for 'real-world' problems such as detecting objects against a cluttered background.

One problem is being able to adapt the feature extraction process to suit the type of structure that predominates in an image, such as lines or textured regions. Although the pixel-pair features were able to vary in terms of the type of light-dark patterns they represented and the spatial relations between the constituent pixels, because of their random nature, there was no provision for changing these characteristics in response to varying image conditions.

Another problem would be trying to achieve a multilevel representation through combinations of these highly generic constructs without encountering combinatorial explosion.

The second approach was algorithmic. In the third set of experiments, homogeneous 'light' and 'dark' polygons were generated within densely-sampled rectangular image 'windows', as described in Section 4.2.2, and a 'useful' subset of windows selected as a representation of pedestrian and non-pedestrian images. These features were more 'engineered' than the pixel-pairs, but were better suited to the greyscale pedestrian images. Being larger and local, they could also more readily form potentially useful higher-level constructs.

With a small set of images they were able to discriminate pedestrians and non-pedestrians reasonably well, with improved performance at the level of pairs of neighbouring windows.

In the fourth and fifth sets of experiments, an algorithm that generates heterogeneous polygons was explored in the context of both binary and greyscale images, Section 4.2.3. These were less 'engineered' than the window features, in that their size and shape were not predetermined, however, the binary 2x2s pattern descriptors used to represent them were chosen by the user, as were the 'light' and 'dark' polygon descriptors within the window features, including greyscale variance and horizontal and vertical variances.

All the above features, due to their simplicity, are vulnerable to image transformations and noise and so more exploration of how systems might autonomously abstract appropriate features and describe them in ways that allow good generalization in representing highly variable data across different visual tasks is needed.

Question 4: Is there a way that structure at higher levels can 'emerge' so that the intermediate word problem and the combinatorial and dimensionality problems can be solved automatically?

The work of the thesis suggests that this is the case.

If simplices representing all possible combinations of constructs at a given level were to be formed, there would be a combinatorial explosion and many of the resulting constructs would be likely to be irrelevant, so some way of enabling the relevant higher level structure to 'emerge' so that the system can avoid the 'curse of dimensionality' is needed.

The second set of experiments uses exact construct-matching to classify at the level of the individual construct. In addition, for the simple geometric shapes involved, the number of different types of construct is limited and so several of the curvature constructs are shared among several of the training objects. This in conjunction with a spatial relation that requires that only suitably-connected neighbouring constructs can form higher level structure, makes it possible to determine which individual constructs and higher-level combinations of them are potentially useful classifiers, as illustrated in the Incidence Matrix, Table 5.1. Hence it is unnecessary to consider all possible combinations of low-level constructs as candidates for

building higher-level representations, since under the required spatial relation, all the potential higher-level structure can be made explicit.

The results of these experiments show that classification at the individual construct level is improved over that at the whole object level and that classification making use of information from multiple neighbouring constructs improves performance still further. In addition, the process of converting mixed-category constructs to the single category of their neighbours enables individual 'pseudo' higher-level constructs to 'emerge' as required, thus further reducing the amount of higher-level structure that needs to be considered.

In the third set of experiments, the 'pedestrian' data is very variable, and in addition, the homogeneous polygon window constructs have variable-length descriptions. This means that multilevel structure, that appears significantly often and that has a tendency to occur more frequently in one class than another, is less likely to emerge than with the curvature constructs of the second set of experiments, without the application of some sort of clustering approach to find prototypes.

This problem was addressed using two approaches that did not involve explicit clustering.

Firstly, pairs of windows that satisfied a particular spatial relation were simply treated as emergent constructs at the next representation level and used to reclassify images that had been insufficiently confidently classified at the level of the individual windows.

Secondly, a 'wrapper-based' technique was applied, classifying test examples and then analysing the output, documented for convenience in the form of a Classification Incidence Matrix, to discover which training images were being instantiated in 'successful' windows (defined as those that were classifying correctly more often than not). In particular, the technique was applied to finding successful pairs of windows under the same spatial constraint as in the first approach, 'emerging' as a result of the classification process, as a joint instantiation in a single training image.

Both approaches brought about a slight improvement in classification performance over that at the lower level, with the wrapper approach performing better, which indicates that manageable quantities of potentially useful higher-level constructs can emerge under suitable constraints, in this case, fixing the number of lower-level constructs to be combined and applying a specific spatial relation.

The first approach was also employed in the fourth and fifth sets of experiments at multiple levels, and in each case, classification tended to improve above the level of the individual construct and the number of constructs was decreased at each successive level.

6.2 Contributions of the thesis

The thesis has made five contributions in the field of building multilevel, multidimensional representations of visual objects, through the application of *hypernetworks* theory.

1. The thesis has demonstrated that the 'fundamental principle of *hypernetworks*' can be applied to form multilevel representations in a variety of visual tasks. The resulting representations have been shown to be useful for object classification tasks ranging from discrimination of simple geometric shapes, to pedestrian recognition and classification of hand-written numerals, thus strengthening the hypothesis that a *hypernetwork*-based representation provides a general architecture for tackling object representation and recognition problems.

2. The thesis has introduced a novel application of an Incidence Matrix to make explicit, at multiple levels of representation, structure that is potentially useful for discriminating object classes. The objects and their constituent constructs are arranged in the rows and columns, respectively, of the matrix, so that spatially-connected subsets of constructs, that co-occur in multiple objects, 'emerge' through the appearance of *maximal rectangles* of '1s' in the body of the matrix. These rectangles can highlight structure

that is shared by objects of the same class, thus illustrating the potential for 'intermediate words' to emerge automatically within a multilevel *hypernetwork*-based representation.

3. The thesis has presented a novel classification heuristic and tested it on the geometric shapes of the second set of experiments. For a test object, knowledge about the classification of its neighbouring constructs is used to 'convert' a given construct of mixed-category to a single-category, either invoking higher-level structure that already exists in the training data, or forming a new combination of constructs that has not yet appeared in any of the training objects and could potentially be a new hub to add to the training set. This construct conversion approach, as well as improving performance over the individual construct level, also addresses the issues of controlling combinatorial explosion and high dimensionality by only using 'selected' higher-level structure as required, the process of seeking higher-level constructs only being initiated by the occurrence of classification conflict at the individual construct level.

4. The thesis has generalized the concept of 'star-hub' analysis, so that, instead of a 'hub' construct having a 'star' of objects in which it appears always in exactly the same form, a hub can now have a star of objects in which its description can vary, while still having some degree of similarity. This allows representations in which constructs are unlikely to match exactly due to the high variability of the data, to be expressed within a *hypernetwork* framework. The idea is illustrated in the third set of experiments, where the 'window' features have variable-length descriptions and each window is considered as a set of 'hubs', each of which has, in its star, all the training images (of both classes) for which it has the same length of description. A test window can only be compared with a hub window that is in the appropriate image location, and whose instantiations in the training images in its star give it the same length of description as the test window. The classification information obtained from matching a set of test objects to the

training images can be analysed to reveal potentially useful structure at multiple representation levels, using the approach described below.

5. The thesis has also introduced an adaptation of the Incidence Matrix, termed the Classification Incidence Matrix, which employs a *hypernetwork* representation in a novel wrapper-based multilevel feature selection method. The Classification Incidence Matrix makes explicit the classification output from the generalized 'star-hub' matching scheme of the previous contribution, in response to a set of test examples. It differs from the standard Incidence Matrix in that, instead of 1s and 0s appearing in the body of the Matrix to indicate the presence or absence, respectively, of a feature in a given object, the body of the Matrix contains the identity of the training image in which a given feature was instantiated when a particular test example was classified. The classification output information for a particular feature, or subset of features can be 'read off' from the Matrix, and used as an aid to feature selection. Feature selection is effected on the basis that, if during classification of a set of test objects, a particular feature is instantiated, more often than not, in a training image of the same class as the test object, that feature is considered to be 'useful' and can be added to the training set. In the third set of experiments, the Classification Incidence Matrix is used for discovering higher-level structure in the form of pairs of overlapping windows. Such structure 'emerges' in the Matrix when a pair of windows, in an appropriate spatial configuration, agrees on the classification of a test object. Pairs of windows found to be 'useful' by the above criterion can then be employed to resolve classification conflict at the individual window level.

The thesis has also contributed to feature selection in high dimensions.

- A modification of the Relief Algorithm was introduced, in the task of feature selection, to enable it to be used in relatively high-dimensional representations. Since the original version relies on determining the distance between exemplars in the feature space, the

'curse of dimensionality' could make comparison of such distances meaningless, and so the modified algorithm only measures the distance between exemplars on the basis of the single feature currently being evaluated. This loses the benefit of distance in the context of all the features, but ensures that the examples being compared are actually close on that single feature. Also, the two-class form of the algorithm was adapted so that instead of using just one nearest hit and one nearest miss to evaluate a feature, it uses k nearest hits and misses, $k > 2$, to reduce the adverse effects of noisy data.

6.3 Suggestions for further work

The work of the thesis has indicated some possible approaches to tackling the issues raised in the four research questions. Furthermore, the results of that work have prompted several further questions, in particular with regard to systems being self-forming and making use of emergent structure to adapt their architecture to task demands, while avoiding combinatorial explosion and dimensionality problems.

6.3.1 Extending the role of the Classification Incidence Matrix in learning multilevel representations

In the second and third sets of experiments, the Incidence Matrix information was only used to form structure at a *higher* level than that of the individual construct. In the third set of experiments, feature selection in the form of the Relief algorithm was applied to select a 'useful' set of individual window constructs, which leads to the question of whether the Classification Incidence Matrix could be used for the selection of these features. Individual windows could be selected on the basis of their ability to classify correctly more often than not, as was done with the higher-level constructs. It would be interesting to discover how the performance of features selected in this way would compare with that of the set selected using Relief.

A related question is whether the Classification Incidence Matrix could be used to reveal structure at levels *higher* than pairs of windows. For example, triples of windows could be

spatially constrained to be immediate neighbours, in a similar way to the window pairs, and the 'useful' triples-structure abstracted in a similar way to that of the pairs, as described in Section 5.6.6. For example, in the Classification Incidence Matrix in Appendix D, the triple comprised of windows 163, 164 and 173, in rows 31 – 33, classifies correctly twelve times and incorrectly just once across the 200-strong test set. So although the triple does not occur frequently, it tends to be a relatively reliable classifier.

The trigger for the system to attempt classification using the window-triples data could be failure to reach a suitable classification confidence threshold for the object at the window-pairs level. Other higher-level combinations could be explored and also, inclusion of less-locally constrained constructs at various levels could be considered as a possible approach to making the higher level structure less sparse.

Only two-class problems have been investigated using star-hub analysis in this work. In multi-class problems, as well as the need for reliable descriptors for features, there is a greater likelihood that at least some individual features will be shared among several classes, and so higher-level representations become even more important for a single multi-class classifier to be able to separate out the classes sufficiently well. An important question is whether a Classification Incidence Matrix can help reveal potentially useful constructs at multiple levels for multiple classes. In this case, it would be rare to find an individual construct, even at higher levels, specific to a single class, so the approach could be 'one-against-all', in the sense that a construct that was selective for a particular class to a greater extent than any of the others would be considered a 'useful' feature. Alternatively, a construct might be selective for a subset of the classes. Another interesting question is how independent a system could or should be in deciding which potentially useful features to select.

There is also the problem, as discussed in Chapter 4, Section 4.5.7 and Chapter 3, Section 3.6, of adapting to changing task requirements, such as learning a new class of object, without having to build a new representation from scratch, and with possibly only a few examples.

The question arises of how a system might make use of existing, possibly shared, 'hub' features to guide the generation of new shared features that include the new class in their 'star'.

6.3.2 Emergent multilevel structure through exact matching of descriptors and constructs

A new approach to the representation of objects and to measuring the similarity between them is developed in Johnson, (In Press). The idea is for a system to be able to generate subsets of useful descriptors for objects or features, while avoiding the potential 'chalk and cheese' problem of measuring distances between entities in a Euclidean space, Section 4.4.

This is achieved by representing objects in terms of descriptors, the exact values of which are set as the vertices of descriptor simplices. Thus, similarity between entities can be measured precisely in terms of the number of descriptor-simplex vertices they have in common.

The effect of this is to cluster the data, and discovery of which subsets of descriptor vertices are forming significant clusters of the entities they represent is made by examining the connectivity of the simplices through the application of 'Q-analysis'.

Q-analysis searches for all the combinations of descriptor values that can occur in representing the features and thus finds all the entities that are q-near or equivalently that share q + 1 descriptor vertices, Section 4.4.1, for all pertinent values of q.

In Johnson, (In Press), the approach is investigated using a small dataset of hand-drawn face shapes, divided equally between 'smiley-faces' and 'frowny-faces', Figure 6.1.

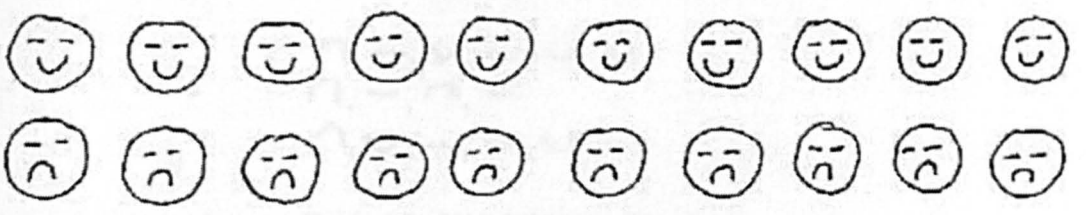


Figure 6.1: Sample from the dataset of 'smiley-faces' and 'frowny-faces'
from Johnson, (In Press), Figure 1.1

The faces are detected by bottom-up aggregation of pixels to form runs and then runs to form polygons, as described in Section 5.3.1.1 of Chapter 5 of this thesis. Finally, at the top level, the

faces are formed from sets of polygons that satisfy a spatial relation that 'binds' polygons that are sufficiently close to one another.

The faces as whole objects cannot readily be discriminated, since they are all roughly the same size and are comprised of the same number of components, so classification is first attempted at the level of the individual polygons. A polygon is defined by four descriptor types: length, height, number of light pixels and number of dark pixels within its bounding box and for each polygon, the system forms a simplex in which each of the four vertices represents the exact value of one of the descriptors.

Q-analysis then reveals the connectivity of the simplices in terms of the number of vertices they share. At different levels of connectivity, clusters of polygons emerge. The system then asks the user if the polygons in a given cluster are 'the same' with respect to the current visual task. If the user's response is 'yes', then the system can create and name a class of polygon. If the answer is 'no', this suggests that the descriptor values represented by the shared vertices are not good for discriminating the polygon types in question, and that the associated vertices could be pruned from the descriptor set, thus tackling the problem of the 'curse of dimensionality'.

At q = 3, at which level the polygons in a cluster must match all four descriptor values exactly, some members of the 'eyes' class are correctly identified. At q = 1, a larger set of eyes emerges, of which those detected at q = 3 are a subset. Also at q = 1, a small 'mixture' of 'smiles' and 'frowns' is detected, while at q = 0, the 'circle' shapes of the heads are correctly identified and three mixed cluster of smiles, frowns and eyes emerge, Figure 6.2.

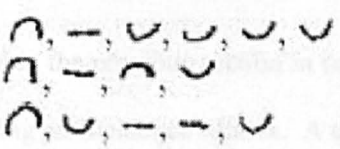


Figure 6.2: Mixed clusters at q = 0
from Johnson, (In Press)

The problem with the mix of 'eyes', 'smiles' and 'frowns' at q = 0 is that the polygons have the same lengths, that is they share a vertex at length = 17, 18 or 19 pixels. Hence to be able to separate these classes, the system needs to introduce a new type of descriptor.

Thus by a cyclic process of trying certain types of descriptor for forming clusters, asking the user whether the clusters are 'useful' for the current purposes, pruning irrelevant and unreliable vertices from the representation and introducing new descriptor types to improve discrimination, a system can build an efficient representation that avoids problems of high dimensionality. In addition, clusters can be expanded by applying 'dilation' Q-analysis, in which for a given descriptor, polygons that match on neighbouring vertex values, say k, k - 1 and k + 1, are merged to form a single cluster.

The work prompts a number of research questions.

- Can this approach be applied to larger datasets of 'real-world' data?

It would be worthwhile to explore whether this approach could be applied to 'real-world' data, such as the MNIST numerals and NiSIS pedestrians data. A simple representation could be applied initially to the data, involving going straight from the descriptor level to that of the whole object, in the case of the numerals. So, instead of having an intermediate representation layer of polygons, the numerals themselves would be represented by a set of descriptors and would be clustered by the vertices of the descriptor simplices to discover which descriptor values were separating the classes well.

- How would such a system cope with multiple classes?

This question was raised in the previous section in connection with the related problem finding higher-level structure using an Incidence Matrix. A difficulty could be that certain descriptors may share the same values for objects of different classes, thus producing mixed-category clusters that it might not be possible or practical to try to separate out by introducing new types of descriptor. In such cases, it may be useful to look at how combinations of 'mixed-category' descriptors, able to collectively separate out the classes, might emerge through Q-analysis.

- How would the system decide when to stop trying new types of descriptor and work with the existing 'mixed-category' descriptors instead?

Random and heuristic approaches to deciding how to adapt the representation could be explored.

- How might Q-analysis discover suitable combinations of 'mixed-category' descriptors values?

One approach might be to look at various levels of connectivity, for combinations of descriptors values that collectively have fewer different categories of object in their clusters than in the clusters of the individual descriptor values. Another problem that is likely to be encountered when clustering noisy data is that many small clusters are formed.

- How can the dilation technique described above be used effectively to merge 'nearby' clusters in a way that minimizes the formation of mixed-category clusters?

One way that could be explored is to randomly select a range of vertex values centred on the value of a candidate vertex for merging, and to merge any vertices that have values that lie within that range. Another approach to the problem of mixed clusters is to introduce intermediate levels of representation, between the 'whole-object' and descriptor levels. This could be explored through the reintroduction of the polygon layer in the representation of the MNIST data.

- Can Q-analysis find a suitable intermediate-level polygonal representation?

A possible broad approach, using the MNIST data to explore the problem, could be to build a three-level system, the lowest level of which would be, say, the sixteen 2x2s descriptors introduced in Section 4.2.3, describing a set of heterogeneous polygons generated using all the numeral classes at the intermediate level, with the whole numerals at the highest level.

Each polygon would be represented by a simplex with sixteen vertices, each of which stores the number of occurrences of its associated 2x2 pattern. Q-analysis would then be applied to cluster the polygons and the user would be asked about the homogeneity of the polygon clusters. Assuming the user could provide the necessary information, the next stage would be to weed out

the vertices that are not good discriminators and then, at the next level, to attempt to cluster the numerals in terms of the polygons that are now being represented by the reduced descriptor set.

For this to be possible, each polygon would have to be labelled, perhaps in a way that relates it to the clusters in which it has appeared, and polygons that end up with the same description would share a vertex on the 'polygon simplex', with each different polygon type having its own vertex.

Once the numerals have been clustered on the basis of the polygon vertices, and Q-analysis applied, the numeral class labels would indicate which of the clusters, if any, contained only numerals of the one class.

It is certain that many interesting research questions would arise during any attempt to implement the above broad approach, including how to manage potential combinatorial and dimensionality problems especially when using polygons to cluster the numerals, but a problem that might arise before dimensionality becomes an issue is:

- What if the user does not know whether the constructs in a particular cluster should be considered as being the same?

This is really the 'intermediate word' problem put another way. In the case of clustered heterogeneous polygons, for example, it may be difficult for the user to know what criteria to employ to decide whether the polygons should be considered to be the same. Also, with a large number of sizeable clusters of constructs that are not readily identifiable by the user, the task of deciding whether they have been reliably clustered could be an onerous one. This suggests that the system might have to complete its representation and test it in an object recognition task before its usefulness could be assessed.

Another point is that a system may be able 'inform' the user about patterns of similarity in data of which the user would otherwise by unaware – patterns of spatial configurations rather than appearance, perhaps, for example in diagnostic images in medicine or industry.

References

Agarwal, S., Awan, A., Roth, D. (2004) Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, **26**, 11.

Agarwal, A. & Triggs, B. (2006) Hyperfeatures – multilevel local coding for visual recognition. Leonardis, A., Bischof, H., & Pinz, A. (Eds): *ECCV 2006, Part 1, LNCS 3951*, 30 – 43.

Arifin, A. Z. & Asano, A. (2006) Image segmentation by histogram thresholding using hierarchical cluster analysis. Pattern *Recognition Letters*, **27**, 1515 – 1521.

Baddeley, R., Abbott, L.F., Booth, M.C.A., Sengpiel, F., Freeman, T., Wakeman, E.A. & Rolls, E.T. (1997) Responses of neurons in primary and inferior temporal visual cortices to natural scenes. *Proceedings of the Royal Society London B*, **264**, 1775-1783.

Ballard, D. H. (1981) Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* **13**, 111 – 122.

Barlow, H. B. (1972) Single units and sensation: A neuron doctrine for perceptual psychology? *Perception*, **1**, 371-394.

Barlow, H. B.(2001) Redundancy reduction revisited. *Network: Computational Neural Systems*, **12**, 241-253.

Bart, E. & Ullman, S. (2005) Cross-generalization: learning novel classes from a single example by feature replacement. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 672 – 679.

Belongie, S., Malik, J., Puzicha, J. (2002) Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 24, 509 – 522.

Best, B. (2010) Pulvinar, www.benbest.com/science/anatmind/anatmd5.html

Biederman, I. (1987) Recognition-by-components: A theory of human image understanding. *Psychological Review*, **94**, 115-147.

Biederman, I. & Gerhardstein, P.C. (1993) Recognizing depth-rotated objects: Evidence and conditions for three-dimensional viewpoint invariance. *Journal of Experimental Psychology: Human Perception and Performance*, **19**, 1162-1182.

Bileschi, S., Wolf, L. (2007) Image representations beyond histograms of gradients: The role of Gestalt descriptors. *Proceedings of the 2007 Conference on Computer Vision and Pattern Recognition*, 1 – 8.

Bishop, C. M. (2002) Neural Networks for Pattern Recognition. ISBN 0 19 853864 2 Oxford University Press.

Bishop, C. M. (2006) Pattern Recognition and Machine Learning. ISBN-10: 0-387-31073-8, 2006 Springer Science + Business Media, LLC.

Borenstein, E. & Ullman, S. (2002) Class-specific, top-down segmentation. *Proceedings of the European Conference on Computer Vision*, May 2002, 109 – 112.

Brunelli, R. & Poggio, T. (1993) Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 10, 1042 – 1052.

Bulthoff, H.H., Edelman, S.Y. & Tarr, M.J. (1995) How are three-dimensional objects represented in the brain? *Cerebral Cortex*, **5**, 247-260.

Canny, J. (1986) A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-8**, 6, November 1986.

Capp, M. D. & Picton, P. D. (2000) The optophone; an electronic blind aid. *Engineering Science and Education Journal*, June 2002, 137 – 143.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., Bray, C. (2004) Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004, Report number 2004/010.

Dalal, N. & Triggs, W. (2005) Histograms of oriented gradients for human detection. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, **2**, 886 – 893.

Das, A. & Gilbert, C.D. (1999) Topography of contextual modulations mediated by short-range interactions in primary visual cortex. *Nature*, **399**, 655-661.

Deco, G. & Rolls, E.T. (2004) A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Research*, **44**, 621-642.

Deco, G. & Zihl, J. (2001) Top-down selective visual attention: A neurodynamical approach. *Visual Cognition*, **8**, 119-140.

Desimone, R. (1996) Neural mechanisms for visual memory and their role in attention *Memory: Recording Experience in Cells and Circuits*. Proceedings of the National Academy of Sciences, USA, Irvine, California, pp. 13494–13499.

Desimone, R. & Duncan, J. (1995) Neural machanisms of selective visual attention. *Annual Review of Neuroscience*, **18**, 193-222.

De Valois, R.L. & De Valois, K.K. (1988) *Spatial Vision*. Oxford University Press, New York.

Duda, R. O. & Hart, P. E. (1971) Use of the Hough transformation to detect lines and curves in pictures. *Communications of the Association for Computing Machinery*. January 1972, 11- 15.

Duncan, J. & Humphreys, G.W. (1992) Beyond the search surface: visual search and attentional engagement. *Journal Experimental Psychology: Human Perception & Performance*, **18**, 578-588.

Edelman, S. (1999) *Representation and Recognition in Vision*. MIT Press, Cambridge, Massachusetts London, England.

Edelman, S. & Intrator, N. (2002) *Visual processing of object structure*. MIT Press.

Elliffe, M.C.M., Rolls, E.T. & Stringer, S.M. (2002) Invariant recognition of feature combinations in the visual system. *Biological Cybernetics*, **86**, 59-71.

Encyclopedia Britannica (2010) http://media-2.web.britannica.com/eb-media/43/79543-004-C3F00EE8.jpg

Epshtein, B. & Ullman S. (2005) Feature hierarchies for object classification. *Proceedings of the Tenth International Conference on Computer Vision (ICCV'05)* **1**, 220 - 227.

Fei-Fei, L., Fergus, R., Perona, P. (2007) Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, **106**, 1, 59 – 70.

Fergus, R., Perona, P., Zisserman, A. (2003) Object class recognition by unsupervised scale-invariant learning. *Proceedings of the 2003 Conference on Computer Vision and Pattern Recognition*, **2**, 264 – 271.

Fergus, R., Perona, P., Zisserman, A. (2005) A sparse object category model for efficient learning and exhaustive recognition. *Proceedings of the Conference in Computer Vision and Pattern Recognition (CVPR'05)*, 380 – 387.

Fisher, R., Perkins, S., Walker, A., Wolfart, E. (2010) Feature detectors – Sobel edge detector. http://homepages.inf.ed.ac.uk/rbf/HIPR2/Sobel.htm

Fisher, R., Perkins, S., Walker, A., Wolfart, E. (2010) Feature detectors – Sobel edge detector. http://homepages.inf.ed.ac.uk/rbf/HIPR2/Zeros.htm

Fleck, M.M. (1996) The topology of boundaries. *Artificial Intelligence*, **80**, 1-26.

Foldiak, P. (2002) *Sparse coding in the primate cortex*. MIT Press, Cambridge, Massachussetts.

Forsyth D. A. & Ponce, J. (2003) Computer Vision A Modern Approach. ISBN 0 13 085198 1 Prentice Hall.

Freeman, W. T. & Roth, M. (1995) Orientation histograms for hand gesture recognition. *IEEE International Workshop on Automatic Face and Gesture Recognition, Zurich, June 1995.*

Freund, Y. & Schapire, R. E. (1999) A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14 (5), 771 – 780, September, 1999.

Fukushima, K. (2004) Neocognitron capable of incremental learning. *Neural Networks*, **17**, 37-46.

Gawne, T.J. & Martin, J.M. (2002) Responses of Primate Visual Cortical V4 Neurons to Simultaneously Presented Stimuli. *J Neurophysiol*, **88**, 1128-1135.

Grill-Spector, K. (2003) The neural basis of object perception. *Current Opinion in Neurobiology*, **13**, 159-166.

Guyon, I. (2008) Practical feature selection: from correlation to causality. *In Mining Massive Datasets for Security, IOS Press, 2008.* URL http://eprints.pascal_network.org/archive/00004038/01/PracticalFS.pdf

Guyon, I. and Elisseeff, A. (2003) An introduction to variable and feature selection. *Journal of Machine Learning Research*, **3**, 1157 – 1182.

Harvey Mudd College, California, Biology handout (2005) The optic chiasm and segregation of parvo- and magno-cellular input to left LGN http://fourier.eng.hmc.edu/e180/handouts/retina/node19.html

Hayward, W.G. (1998) Effects of outline shape in object recognition *Journal of Experimental Psychology: Human Perception and Performance*, **24**, 427-440.

Hegde, J. & Van Essen, D.C. (2000) Selectivity for Complex Shapes in Primate Visual Area V2. *Journal of Neuroscience*, **20**, 1- 6.

Helmholtz, 1867, 1925 version, "Treatise on physiological optics", Electronic edition, 2001: University of Pennsylvania, URL: http://psych.upenn.edu/backuslab/helmholtz, Vol III.

Heydt, R.V.D. (2003) *Image parsing mechanisms of the visual cortex*. MIT Press, Cambridge, Massachussetts.

Hochstein, S. & Ahissar M. (2002) View from the top: hierarchies and reverse hierarchies in the visual system. *Neuron*, **36**, 791 – 804.

Hoyer, P.O. & Hyvarinen, A. (2002) A multi-layer sparse coding network learns contour coding from natural images. *Vision Research*, **42**, 1593-1605.

Huang, Y., Huang, K., Tao, D., Wang, L., Tan, T., Li, X. (2008) Enhanced biologically inspired model. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'08)*.

Hubel, D.H. (1995) *Eye, Brain and Vision*. Scientific American Library.

Iravani, P., Johnson, J. H., Rapanotti, L. (2005) Robotics and the Q-analysis of behaviour. *Proceedings of the 10th International Symposium on Artificial Life (AROB)* 4-6 February, 2005, Beneppu, Japan.

Itti, L. & Koch, C. (2001) Feature combination strategies for saliency-based visual attention systems. *J. Electronic Imaging , Jan P. Allebach; Ed.*, **10**, 161-169.

James, W. (1890) *The Principles of Psychology*, psychclassics.yorku.ca/James/Principles/index.htm Ch 11, 6

Janecek, A. G. K., Gansterer, W. N., Demel, M. A., Ecker, G. F. (2008) On the relationship between feature selection and classification accuracy. *Journal of Machine Learning Research: Workshop and Conference Proceedings* **4**: 90 – 105.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y. (2009) What is the best multi-stage architecture for object recognition? *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

Jarosz, Q., 2010, Structure of a typical neuron. http://en.wikipedia.org/wiki/Neuron

Johnson, J. H. (2006) Hypernetworks for reconstructing the dynamics of multilevel systems. *Proceedings of the European Conference on Complex Systems*, Oxford University, 25th -29th September 2006.

Johnson, J. H. (2007) DIOCLES: modelling the brain by a polynode hypernetwork. The Open University, UK. 22nd April, 2007.

Johnson, J. H. (In Press) *Hypernetworks in the Science of Complex Systems*. Imperial College Press, ISBN-10: 186094972X

Johnson, J. H. & Simon, J. (2001) Fundamental structures for the design of machine vision systems. *Mathematical Geology*, **33**, 331 – 352.

Johnson, J. H. & Sugisaka, M. (2006) Construct abstraction for automatic information abstraction from digital images. *Final Report, 15th December, 2004 – 1st May, 2006*, Accession Number ADA455945.

Jurie, F. & Triggs, W. (2005) Creating efficient codebooks for visual recognition. *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, **1**, 604 – 610.

Kadir, T. & Brady, J. M. (2001) Scale, saliency and image description. *International Journal of Computer Vision*, **45**, 2, 83 – 105.

Kastner, S. & Pinsk, M.A. (2004) Visual attention as a multilevel selection process. *Cognitive, Affective, & Bahavioral Neuroscience*, **4**, 483-500.

Kersten, D., Mamassian, P. & Yuille, A. (2004) Object perception as Bayesian inference. *Annual Review of Psychology*, **55**, 271-304.

Keysers, D., Deselaers, T., Ney, H. (2004) Pixel-to-pixel for image recognition using Hungarian Graph Matching. *Proceedings of DAGM (German Association for Pattern Recognition) Symposium on Pattern Recognition*, 154 – 162.

Kimball, J. W. (2010) The Compound Eye
http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/C/CompoundEye.html

Kira, K. & Rendell, L. A. (1992) The feature selection problem: traditional methods and a new algorithm. *Proceedings of the Tenth National Conference on Artificial Intelligence*, 129 – 134. Menlo Park: AAAI Press/The MIT Press, 1992.

Koch, C. (2004) Selective Visual Attention and Computational Models. *CNS/Bi 186: Attention.*

Kolb, H., Fernandez, E., Nelson, R. (2010) The Organization of the Retina and Visual System
http://webvision.med.utah.edu/sretina.html

Kosslyn, S.M. (1994) *Image and Brain: Resolving the Imagery Debate*. MIT Press, Cambridge, Massachusetts, London, England.

Kpalma, K. & Ronsin, J. (2006), Multiscale contour description for pattern recognition. *Pattern Recognition Letters*, **27**, 1545 – 1559.

Lai, J. H., Yuen, P. C., Feng, G. C. (2001) Face recognition using holistic Fourier invariant features. *Pattern Recognition*, **34**, 95 – 109.

Lee, T.S. (2003) Computations in the early visual cortex. *Journal of Physiology-Paris*, **97**, 121-139.

Lee, T.S. & Mumford, D. (2003) Hierarchical Bayesian inference in the visual cortex. *Journal of the Optical Society of America*, **20**, 1434-1448.

Lee, T.S., Mumford, D., Romero, R. & Lamme, V.A.F. (1998) The role of the primary visual cortex in higher level vision. *Vision Research*, **38**, 2429-2454.

Lecun, Y. & Cortes, C. (2010) The MNIST Database. http://yann.lecun.com/exdb/mnist/

Lecun, Y., Haffner, P., Bottou, L., Bengio, J. (1999) Object recognition with gradient-based learning. in *Feature Grouping* (D.Forsyth, ed), 28 pages.

Lehky, S.R., Sejnowski, T.J. & Desimone, R. (2005) Selectivity and sparseness in the responses of striate complex cells. *Vision Research*, **45**, 57-73.

Lennie, P. (2003) The Cost of Cortical Computation. *Current Biology*, **13**, 493-497.

Lerner, Y., Hendler, T. & Malach, R. (2002) Object-completion Effects in the Human Lateral Occipital Complex. *Cerebral Cortex*, **12**, 163-177.

Levi, D. & Ullman, S. (2010) Learning to classify by ongoing feature selection. *Image and Vision Computing*, **28**, 715 – 723.

Liu, H., Lu, H., Yu, L. (2003) Active sampling: an effective approach to feature selection. SIAM *International Conference on Data Mining*, San Francisco, May 1 – 3, 2003, Poster Presentation.

Logothetis, N.K. & Sheinberg, D.L. (1996) Visual Object Recognition. *Annual Review of Neuroscience*, **19**, 577-621.

Lowe, D. G. (1999) Object recognition from local scale-invariant features. *Proceedings of the Seventh International Conference on Computer Vision*, **2**, 1150 – 1157.

Maree, R., Geurts, P., Piater, J., Wehenkel, L. (2005) Random subwindows for robust image classification. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 34 – 40.

Marr, D. (1982) *Vision: a computational investigation into the human representation and processing of visual information.* W H Freeman, SanFrancisco, CA.

Martinez, A. M & Kak, A. C. (2001) PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**, 2, 228 – 233.

Martinez-Munoz, G., Zhang, W., Payet, N., Todorovic, S. (2008) Dictionary-free categorization of very similar objects via stacked evidence trees.

McGill, (2010) Canadian Institute of Neurosciences
www.thebrain.mcgill.ca/flash/d/d_02/d_02_cl/d_02_cl_vis/d_02_cl_vis.html

McLeod, P., Driver, J. & Crisp, J. (1988) Visual search for a conjunction of movement and form is parallel. *Nature*, **332**, 154-155.

Mel, B. W. (1997) SEEMORE: Combining color, shape and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Computation*, **9**, 777 – 804.

Mel, B. W. & Fiser, J. (2000) Minimizing binding errors using learned conjunctive features. *Neural Computation*, **12**.

Milner, D. A. & Goodale, M. A. (1998) The visual brain in action. *Psyche*, **4** (12) October 1998.

Mitchell, T. M. (1997) Machine Learning. McGraw-Hill publishers.

Moosmann, F. Triggs, W., Jurie, F. (2007) Fast discriminative visual codebooks using randomized clustering forests. *Advances in Neural Information Processing Systems*, **19**, 985 - 992.

Munder, S. & Gavrila, D. M. (2006) An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 11, November 2006.

Murphy, K., Torralba, A., Freeman, W. T. (2003) Using the forest to see the trees: A graphical model relating features, objects and scenes.

Murray, S.O., Schrater, P. & Kersten, D. (2004) Perceptual grouping and the interactions between visual cortical areas. *Neural Networks*
Vision and Brain, **17**, 695-705.

Mutch, J. & Lowe, D. G. (2006) Multiclass object recognition with sparse, localized features. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 11 – 18.

Nilsson D. E. & Pelger, S. (1994) A pessimistic estimate of the time required for an eye to evolve. *Proc. Royal Soc. London B*, **256**, 53-58.

Oliva, A. & Torralba, A. (2006) Building the gist of a scene: the role of global image features in recognition. In *Martinez-Conde, Macknik, Martinez, Alonso & Tse (Eds) Progress in Brain Research*. **155**, 23 – 36.

Olshausen, B. A., Anderson, C. & Van Essen, D. (1993) A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *J. Neurosci.*, **13**, 4700-4719.

Olshaushen B. A. in Chalupa, L.M. & Werner, J.S. (Eds), (2003) Principles of image representation in visual cortex. *The Visual Neurosciences*. MIT Press, pp. 1603-1615.

Olshausen, B. A. & Field, D. J. (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, **381**, 607-609.

Olshausen, B.A. & Field, D.J. (2005) How close are we to understanding V1? *Neural Computation*, **17**, 1665-1699.

Opelt, A., Pinz, A., Zisserman, A. (2006) A boundary fragment model for object detection. *Proceedings of the IEEE European Conference on Computer Vision*, **2**, 575 – 588.

Opelt, A. & Pinz, A. (2006) Incremental learning of object detectors using a visual shape alphabet. *Proceedings of the 2006 Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 3 – 10.

Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Poggio, T. (1997) Pedestrian detection using wavelet templates. *Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition*, 193 – 197.

Palmer, S.E. (1999) *Vision Science: Photons to Phenomenology*. MIT Press, Cambridc, Massachusetts, London, England.

Pasupathy, A. & Connor, C.E. (1999) Responses to Contour Features in Macaque Area V4. *J Neurophysiology*, **82**, 2490-2502.

Pasupathy, A. & Connor, C.E. (2001) Shape Representation in Area V4: Position-Specific Tuning for Boundary Conformation *J Neurophysiology*, **86**, 2505-2519.

Perrett, D.I., Oram, M.W. & Ashbridge, E. (1998) Evidence accumulation in cell populations responsive to faces: an account of generalisation of recognition without mental transformations. *Cognition*, **67**, 111-145.

Pham, T. V. & Smeulders, A. W. M. (2006) Learning spatial relations in object recognition. *Pattern Recognition Letters*, **27**, 1673 – 1684.

Picton, P. D. (1994) *Introduction to Neural Networks*. MacMillan Press Ltd.

Picton, P. D. & Capp M. D. (2008) Relaying scene information to the blind via sound using cartoon depth maps. *Image and Vision Computing* **26** (2008) 570 – 577.

Pinto, N., Cox, D. D., DiCarlo, J. J. (2008) Why is real-world visual object recognition hard? *Public Library of Science (PLoS) Computational Biology*, **4**, 1, e27.

Pollen, D.A., Przybyszewski, A.W., Rubin, M.A. & Foote, W. (2002) Spatial Receptive Field Organization of Macaque V4 Neurons. *Cereb. Cortex*, **12**, 601-616.

Quattoni, A., Collins M., Derrell, T. (2008) Transfer learning for image classification with sparse prototype representations. *Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1 – 8.

Ranzato, M., Poultney, C., Chopra, S., LeCun, Y. (2006) Efficient learning of sparse representations with an energy-based model. In J. Platt et al (Eds) *Advances in Neural Information Processing Systems (NIPS, 2006)*, MIT Press.

Rao, R.P.N. & Ballard, D.H. (1999) Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, **2**, 79-87.

Reynolds, J.H., Chelazzi, L. & Desimone, R. (1999) Competitive Mechanisms Subserve Attention in Macaque Areas V2 and V4. *The Journal of Neuroscience*, **19**, 1736-1753.

Riesenhuber, M. & Poggio, T. (1999a) Hierarchical models of object recognition in cortex. *Nature neuroscience*, **2**, 1019-1025.

Riesenhuber, M. & Poggio, T. (1999b) Are cortical models really bound by the "binding problem"? *Neuron*, **24**, 87-93.

Riesenhuber, M. & Poggio, T. (2002) Neural mechanisms of object recognition. *Current Opinion in Neurobiology*, **12**, 162-168.

Robnik-Sikonja, M & Kononenko, I.. (2003) Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, **53**, 23 – 69.

Rolls, E.T., Aggelopoulos, N.C., Franco, L. & Treves, A. (2004) Information encoding in the inferior temporal visual cortex: contributions of the .ring rates and the correlations between the .ring of neurons. *Biological Cybernetics*, **90**, 19-32.

Rolls, E.T. & Deco, G. (2002) *Computational Neuroscience of Vision*. Oxford University Press.

Rolls, E.T. & Milward, T. (2000) A model of invariant object recognition in the visual system: learning rules, activation functions, lateral inhibition, and information-based performance measures. *Neural Computation*, **12**, 2547-2572.

Rolls, E.T. & Treves, A. (1998) *Neural Networks and Brain Function*. Oxford University Press.

Rose, V. 2009, *Autonomous construct generation for multilevel representation and recognition of hand-written numerals*, Technical Report, 8/11/09

Rose, V. & Johnson, J. H. (2005) Shape-recognition using randomly selected pixel pair neurons. In M. Sugisaka (ed) *Proceedings of Artificial Life and Robotics 10.*

Rousselet, G.A., Thorpe, S.J. & Fabre-Thorpe, M. (2004) How parallel is visual processing in the ventral pathway? *Trends in Cognitive Sciences*, **8**, 363-370.

Rzevski, G. (1995), (Ed) Mechatronics: Designing Intelligent Machines, Volume 1: Perception, Cognition and Execution. Butterworth-Heinemann Ltd in association with The Open University.

Satoh, S., Kuroiwa, J., Aso, H. & Miyake, S. (1997) Recognition of rotated patterns using neocognitron. *Proceedings of the International Conference on Neural Information Processing*, **1**, 112-116.

Scholkopf, B. (2001) The kernel trick for distances. *Advances in Neural Information Processing Systems*, **13**, MIT Press.

Serre, T., Wolf, L. & Poggio, T. (2005) Object recognition with features inspired by visual cortex. *CVPR05 Proceedings*, 994 – 1000.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T. (2006) Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern analysis and Machine Intelligence.* **20**, 20.

Shepard, R. N. (1964). Attention and the metric structure of the stimulus space. *Journal of Mathematical Psychology*, **1**, 54-87.

Shi, J & Malik, J. (2000) Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (8) 731 – 737.

Siedlecki, W. & Sklansky, J. (1993) On automatic feature selection. In Chen, C. H., Pau, L. F., Wang, P.S.P. (eds) *Handbook of Pattern Recognition and Computer Vision*, World Scientific Publishing Co. Pte. Ltd., ISBN 981-02-1136-8.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., Freeman, W. T. (2005) Discovering objects and their location in images. *Proceedings of the Tenth International Conference on Computer Vision*, **1**, 370 – 377.

Shotton, J., Blake, A. & Cipolla., R. (2008) Multi-scale categorical object recognition using contour fragments. *IEEE Transactions of Pattern Analysis and Machine Learning*, **30, (7)**, 1270 – 1281.

Shotton, J., Winn, J., Rother, C., Criminisi, A. (2009) TextonBoost for image understanding: multi-class object recognition and segmentation by jointly modelling texture, layout and context. *International Journal of Computer Vision*, **81**, (1), 2 – 23.

Soegaard, M. (2010). *Encyclopedia entry on Gestalt principles of form perception.* Retrieved 16 April 2010 from Interaction-Design.org: http://www.interaction-design.org/encyclopedia/gestalt_principles_of_form_perception.html

Tanaka, K. (2003) Columns for Complex Visual Object Features in the Inferotemporal Cortex: Clustering of Cells with Similar but Slightly Different Stimulus Selectivities *Cerebral Cortex*, **13**, 90-99.

Tarr, M.J. & Bulthoff, H.H. (1998) Image-based object recognition in man, monkey and machine. *Cognition*, **67**, 1-20.

Tarr, M.J. & Cheng, Y.D. (2003) Learning to see faces and objects. *Trends in Cognitive Sciences*, **7**, 23-30.

Thrun, S. (1996) Is learning the *n*-th thing any easier than learning the first? Advances *in Neural Information Processing Systems. (NIPS)* **8**, 640 – 646, MIT Press.

Torralba, A., Murphy, K. P., Freeman, W. T. (2007) Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**, 5, 854 – 869.

Treisman, A. (1988) Features and objects: The fourteenth Bartlett Memorial Lecture. *Quarterly Journal of Experimental Psychology*, **40A**, 201-237.

Treisman, A. & Gelade, G. (1980) A feature integration theory of attention. *Cognitive Psychology*, **12**, 97-136.

Tu, Z. (2007) Learning generative models via discriminative approaches. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1 - 8.

Turk, M. A. & Pentland, A. P. (1991) Face recognition using eigenfaces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1991, 586 – 591.

Tversky, A. (1977). Features of similarity. *Psychological Review, 84*, 327-352.

Ullman, S. & Sali, E. (2000), Object classification using a fragment-based representation. In S.-W Lee, H. H. Bulthoff, T Poggio (Eds): *BMCV*, 2000, LNCS 1811, 73 – 87.

Ullman, S. & Bart, E. (2004) Recognition invariance obtained by extended and invariant features. *Neural Networks*, **17**, 833 – 848.

Ulusoy, I. & Bishop, I. (2005) Generative versus discriminative methods for object recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2, 02, 258 – 265.

University of Illinois at Urbana Champaign (2005) The response patterns of the retinal ganglion cells, http://soma.npa.uiuc.edu/courses/bio303/Ch11.html

Usher, M. & Niebur, E. (1996) Neurons in visual search: A mechanism for top-down selective attention. *Journal of Cognitive Neuroscience*, **8**, 311-327.

VanRullen, R., Reddy, L. & Fei-Fei, L. (2005) Binding is a local problem for natural objects and scenes. *Vision Research*, **45**, 3133-3144.

Vidal-Naquet, M. & Ullman, S. (2003) Object recognition with informative features and linear classification. *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03)*, 281 – 288.

Vinje, W.E. & Gallant, J.L. (2000) Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, **287**, 1273-1276.

Viola, P. & Jones, M. (2001) Rapid object detection using a boosted cascade of simple features. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* **1**, 511 – 518.

Von der Malsburg, C. (1999) The What and Why of Binding: The Modeler's Perspective. *Neuron*, **24**, 95-101.

Walker, G.A., Ohzawa, I. & Freeman, R.D. (2000) Suppression outside the classical cortical receptive field. *Visual Neuroscience*, **17**, 369-379.

Wallach, H. M. (2004) Conditional random fields: An introduction. University of Pennsylvania CIS Technical Report MS-CIS-04-21.

Wallis, G. & Rolls, E.T. (1997) Invariant face and object recognition in the visual system. *Progress in Neurobiology*, **51**, 167-194.

Weber, M., Welling, M., Perona, P. (2000) Unsupervised learning of models for recognition. *European Conference on Computer Vision, Lecture Notes in Computer Science*, **1842 – 2000**, Springer Berlin/Heidelberg, 18 – 32.

Wickelgren, W. A. (1969) Context-sensitive coding, associative memory, and serial order in (speech) behaviour. Reprinted from *Psychological Review*, **76**, 1, 1 – 15.

Wikipedia_Ocelli (2010) http://en.wikipedia.org/wiki/Ocelli

Wikipedia, Scale invariant feature transform (2010) http://en.wikipedia.org/wiki/Scale-invariant_feature_transform

Wikipedia_Simple_eyes_in_invertebrates, (2010) http://www.en.wikipedia.org/Simple_eyes_in_invertebrates

University of Wisconsin (2010), Superior colliculus, www.neuroanatomy.wisc.edu/Bs97/TEST/P23/ov.hrm

Wolf, L., Bileschi, S., Meyers, E. (2006) Perception strategies in hierarchical vision systems. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, 2153 – 2160.

Wolfe, J.M. (1994) Guided search 2.0 A revised model of visual search. *Psychonomic Bulletin & Review*, **1**, 202-238.

Wolfe, J.M. (1996) Extending guided search: Why Guided Search needs a preattentive "Item Map". In Kramer, A.F., Coles, M.G.H., Logan, G.D. (eds.) *Converging operations in the study of visual attention.* American Psychological Association, Washington, DC, pp. 247-270.

Wolfe, J.M. (2003) Moving towards solutions to some enduring controversies in visual search. *Trends in Cognitive Sciences*, **7**.

Yu, C., Klein, S.A. & Levi, D.M. (2003) Cross- and iso- oriented surrounds modulate the contrast response function: The effect of surround contrast. *Journal of Vision*, **3**, 527-540.

Zhou, H., Friedman, H.S. & von der Heydt, R. (2000) Coding of Border Ownership in Monkey Visual Cortex. *J. Neuroscience.*, **20**, 6594-6611.

Zhou, J., Peng, H., Suen, C. S. (2008) Data-driven decomposition for multi-class classification. *Pattern Recognition*, **41**, 67 – 76.

Zhu, J. & von der Malsburg, C. (2004) Maplets for correspondence-based object recognition. *Neural Networks*, **17**, 1311-1326.

Appendices

Appendix A Randomly-selected pixel pair features - results

12 23 67 7	-55 16	0 30 43 74	-43 -44
18 59 16 3	2 56	60 2 5 24	55 -22
73 8 36 74	37 -66	22 14 15 46	7 -32
45 21 74 58	-29 -37	33 51 1 32	32 19
70 53 72 31	-2 22	38 0 7 31	31 -31
56 70 43 17	13 53	67 68 42 11	25 57
40 56 66 24	-26 32	61 2 74 33	-13 -31
60 28 3 29	57 -1	72 51 56 19	16 32
7 47 48 2	-41 45	9 53 60 65	-51 -12
51 44 40 14	11 30	72 33 37 34	35 -1
36 19 54 16	-18 3	36 4 56 35	-20 -31
12 30 43 20	-31 10	65 10 69 40	-4 -30
23 20 35 13	-12 7	46 34 37 55	9 -21
37 16 18 37	19 -21	66 8 59 74	7 -66
24 54 67 36	-43 18	70 17 54 11	16 6
3 72 47 37	-44 35	38 13 16 1	22 12
72 41 7 24	65 17	11 39 10 15	1 24
25 58 68 53	-43 5	68 6 51 35	17 -29
65 46 71 38	-6 8	50 71 43 7	7 64
16 54 10 10	6 44	0 32 35 10	-35 22
23 14 57 16	-34 -2	28 34 40 53	-12 -19
58 16 66 46	-8 -30	60 41 15 14	45 27
72 54 28 2	44 52	43 35 63 61	-20 -26
10 6 73 74	-63 -68	34 8 42 28	-8 -20
27 1 4 47	23 -46	35 17 69 35	-34 -18
15 30 39 11	-24 19	0 21 1 55	-1 -34
39 33 39 63	0 -30	49 35 53 18	-4 17
71 54 69 3	2 51	73 29 58 14	15 15
45 68 49 19	-4 49	6 50 61 42	-55 8
18 30 61 43	-43 -13	58 36 55 44	3 -8
67 58 65 15	2 43	41 11 20 46	21 -35
10 69 39 61	-29 8	25 8 71 22	-46 -14
33 4 3 36	30 -32	52 22 26 7	26 15
9 35 27 12	-18 23	16 16 17 41	-1 -25
57 42 64 14	-7 28	18 25 66 59	-48 -34
53 9 0 69	53 -60	11 8 26 42	-15 -34
33 36 71 19	-38 17	64 46 11 73	53 -27
40 23 33 66	7 -43	32 66 74 49	-42 17
65 22 52 56	13 -34	43 27 54 7	-11 20
45 34 36 20	9 14	72 21 1 16	71 5
47 16 35 74	12 -58	47 55 47 48	0 7
53 51 27 15	26 36	52 36 33 52	19 -16
54 39 17 21	37 18	18 43 57 59	-39 -16
10 28 32 35	-22 -7	61 0 73 25	-12 -25
17 48 42 33	-25 15	41 64 43 57	-2 7
20 68 20 41	0 27	32 49 43 71	-11 -22
27 53 65 69	-38 -16	21 42 70 69	-49 -27
53 65 3 43	50 22	58 70 30 50	28 20
42 55 53 19	-11 36	47 45 74 11	-27 34
58 6 38 39	20 -33	46 39 37 2	9 37
2 11 16 42	-14 -31	51 58 71 51	-20 7
5 46 47 10	-42 36	27 30 0 19	27 11
42 67 20 21	22 46	49 41 21 9	28 32
41 21 28 11	13 10	46 21 33 60	13 -39
3 46 45 10	-42 36	25 30 8 43	17 -13
45 22 6 24	39 -2	36 34 36 72	0 -38
18 65 38 42	-20 23	7 30 33 17	-26 13
14 39 41 64	-27 -25	5 10 25 13	-20 -3
40 66 23 50	17 16	58 23 67 20	-9 3
34 7 68 72	-34 -65	51 24 24 3	27 21

48 29 59 16	-11 13	68 67 63 49	5 18
40 53 63 37	-23 16	66 60 5 23	61 37
42 58 34 71	8 -13	58 70 32 35	26 35
47 59 1 9	46 50	8 40 0 34	8 6
9 52 21 54	-12 -2	2 12 43 5	-41 7
67 0 11 53	56 -53	24 35 68 72	-44 -37
44 47 7 49	37 -2	1 51 74 74	-73 -23
48 9 59 57	-11 -48	20 22 65 5	-45 17
49 12 49 70	0 -58	24 63 48 73	-24 -10
50 63 6 44	44 19	9 12 57 63	-48 -51
20 14 69 26	-49 -12	69 13 61 21	8 -8
30 44 20 9	10 35	24 3 35 56	-11 -53
49 52 38 64	11 -12	74 32 37 5	37 27
31 42 25 67	6 -25	34 29 63 52	-29 -23
19 16 22 64	-3 -48	55 32 41 50	14 -18
11 18 15 10	-4 8	11 7 69 24	-58 -17
70 37 1 15	69 22	33 36 41 61	-8 -25
21 44 64 51	-43 -7	57 51 63 14	-6 37
28 39 16 24	12 15	9 43 6 65	3 -22
22 27 63 48	-41 -21	22 3 74 66	-52 -63
19 30 0 26	19 4	62 60 66 24	-4 36
37 10 28 12	9 -2	14 17 25 60	-11 -43
10 61 23 14	-13 47	12 73 20 13	-8 60
5 60 29 36	-24 24	32 2 6 48	26 -46
49 24 60 18	-11 6	47 25 57 57	-10 -32
65 60 28 45	37 15	16 44 62 20	-46 24
9 16 20 3	-11 13	50 21 42 41	8 -20
8 73 22 73	-14 0	43 68 73 13	-30 55
25 25 50 40	-25 -15	4 45 23 68	-19 -23
38 17 34 18	4 -1	35 28 54 10	-19 18
59 73 34 16	25 57	20 63 30 53	-10 10
9 73 45 58	-36 15	37 22 53 42	-16 -20
42 4 44 66	-2 -62	37 7 24 55	13 -48
16 14 56 26	-40 -12	47 24 41 19	6 5
3 22 67 54	-64 -32	30 73 31 68	-1 5
43 34 54 56	-11 -22	7 59 31 25	-24 34
17 10 19 56	-2 -46	66 26 8 34	58 -8
46 62 23 64	23 -2	47 42 62 65	-15 -23
24 32 10 40	14 -8	44 64 31 45	13 19
11 74 64 50	-53 24	28 13 12 72	16 -59
29 69 74 1	-45 68	1 46 21 38	-20 8
19 71 69 1	-50 70	33 20 10 4	23 16
41 63 23 19	18 44	48 51 57 61	-9 -10
46 72 21 34	25 38	28 12 60 72	-32 -60
24 24 23 32	1 -8	21 36 70 36	-49 0
1 43 24 25	-23 18	51 59 60 33	-9 26
46 0 37 54	9 -54	3 68 51 62	-48 6
18 72 49 66	-31 6	1 31 64 17	-63 14
64 30 38 21	26 9	19 1 8 58	11 -57
7 68 37 67	-30 1	73 0 38 50	35 -50
28 15 58 50	-30 -35	15 13 71 46	-56 -33
53 47 68 31	-15 16	29 5 24 21	5 -16
11 44 61 73	-50 -29	23 55 37 47	-14 8
42 5 22 12	20 -7	41 9 28 49	13 -40
57 56 2 14	55 42	25 59 61 46	-36 13
12 56 41 28	-29 28	1 37 48 52	-47 -15
67 5 5 69	62 -64	49 16 16 60	33 -44
35 53 10 35	25 18	33 10 38 18	-5 -8
48 74 54 37	-6 37	61 17 71 11	-10 6
43 70 46 1	-3 69	52 71 0 14	52 57

74 30 24 33	50 -3	3 44 47 28	-44 16
46 10 48 29	-2 -19	10 48 7 27	3 21
32 14 62 49	-30 -35	51 16 64 18	-13 -2
5 38 67 12	-62 26	19 37 38 32	-19 5
14 7 18 2	-4 5	64 38 0 59	64 -21
62 0 57 18	5 -18	3 15 20 65	-17 -50
72 4 14 50	58 -46	50 19 48 26	2 -7
5 21 8 36	-3 -15	23 21 0 30	23 -9
11 39 31 13	-20 26	67 1 32 38	35 -37
6 29 45 34	-39 -5	37 33 8 61	29 -28
72 39 6 72	66 -33	68 64 70 1	-2 63
18 24 25 34	-7 -10	44 29 5 34	39 -5
29 40 30 32	-1 8	11 41 32 72	-21 -31
43 15 33 23	10 -8	26 7 32 33	-6 -26
65 40 71 66	-6 -26	60 45 11 22	49 23
32 7 46 63	-14 -56	58 69 37 53	21 16
47 37 64 23	-17 14	51 33 16 9	35 24
73 60 12 22	61 38	14 59 63 39	-49 20
7 51 17 73	-10 -22	39 12 10 6	29 6
60 42 66 10	-6 32	19 65 48 22	-29 43
4 7 53 51	-49 -44	35 47 74 19	-39 28
13 36 13 24	0 12	4 23 25 57	-21 -34
41 9 7 1	34 8	5 20 5 22	0 -2
57 68 9 70	48 -2	38 23 58 4	-20 19
3 64 11 15	-8 49	55 73 46 50	9 23
56 36 71 33	-15 3	64 51 57 55	7 -4
55 57 23 59	32 -2	72 15 44 18	28 -3
46 5 28 20	18 -15	15 37 53 27	-38 10
3 10 71 35	-68 -25	11 23 10 33	1 -10
46 22 30 35	16 -13	28 70 10 7	18 63
59 20 38 33	21 -13	53 26 37 50	16 -24
4 12 55 17	-51 -5	62 53 32 34	30 19
35 72 42 3	-7 69	12 14 35 6	-23 8
62 10 66 56	-4 -46	32 24 59 44	-27 -20
33 20 16 11	17 9	65 45 4 63	61 -18
63 47 30 3	33 44	67 17 26 49	41 -32
56 67 46 51	10 16	74 16 41 35	33 -19
2 45 40 4	-38 41	27 46 65 9	-38 37
52 31 51 54	1 -23	5 8 51 8	-46 0
19 7 42 19	-23 -12	19 3 7 70	12 -67
64 49 12 24	52 25	34 53 29 2	5 51
20 7 61 52	-41 -45	44 49 4 58	40 -9
43 29 19 71	24 -42	29 66 49 15	-20 51
15 56 16 22	-1 34	50 24 0 16	50 8
66 50 72 52	-6 -2	19 2 16 67	3 -65
39 56 58 24	-19 32	38 45 41 53	-3 -8
17 51 6 46	11 5	19 45 31 60	-12 -15
47 69 42 8	5 61	30 34 52 62	-22 -28
8 24 70 39	-62 -15	8 20 63 57	-55 -37
39 62 38 12	1 50	12 43 14 27	-2 16
33 13 52 6	-19 7	24 53 68 6	-44 47
1 9 6 4	-5 5	48 52 61 0	-13 52
72 57 23 63	49 -6	29 1 34 31	-5 -30
43 0 60 39	-17 -39	16 2 60 42	-44 -40
68 47 10 22	58 25	70 48 56 40	14 8
9 0 5 53	4 -53	40 41 61 1	-21 40
73 14 45 35	28 -21	17 44 1 27	16 17
67 33 29 62	38 -29	13 69 2 63	11 6
47 72 65 31	-18 41	6 68 10 9	-4 59
29 71 34 21	-5 50	11 18 74 24	-63 -6

52 12 58 5	-6 7	16 6 67 3	-51 3
39 23 40 61	-1 -38	17 30 0 27	17 3
49 47 72 4	-23 43	5 72 6 15	-1 57
72 65 39 59	33 6	2 27 37 34	-35 -7
43 56 59 56	-16 0	29 56 36 33	-7 23
52 62 8 4	44 58	41 57 39 44	2 13
69 68 65 17	4 51	65 1 48 67	17 -66
27 43 64 44	-37 -1	46 61 16 16	30 45
50 58 21 40	29 18	7 11 20 47	-13 -36
55 32 68 44	-13 -12	56 20 28 15	28 5
3 68 10 71	-7 -3	26 5 1 62	25 -57
23 29 22 31	1 -2	4 22 46 43	-42 -21
64 70 28 66	36 4	38 38 24 22	14 16
71 53 20 33	51 20	69 34 13 10	56 24
41 15 28 21	13 -6	45 60 73 61	-28 -1
40 2 58 56	-18 -54	31 63 15 8	16 55
3 33 42 38	-39 -5	5 37 47 47	-42 -10
6 8 70 23	-64 -15	10 29 74 25	-64 4
29 59 68 43	-39 16	73 59 48 64	25 -5
41 8 65 41	-24 -33	35 42 12 3	23 39
65 15 44 7	21 8	69 73 25 72	44 1
12 67 37 58	-25 9	32 73 35 28	-3 45
5 20 28 2	-23 18	50 57 37 55	13 2
13 16 37 55	-24 -39	36 17 21 25	15 -8
18 54 66 3	-48 51	67 6 34 66	33 -60
21 33 22 37	-1 -4	12 74 43 20	-31 54
36 21 40 72	-4 -51	15 45 33 41	-18 4
69 8 40 70	29 -62	49 54 31 52	18 2
22 48 48 72	-26 -24	42 0 64 37	-22 -37
40 44 25 68	15 -24	74 20 32 45	42 -25
29 10 6 33	23 -23	50 37 17 44	33 -7
66 35 59 26	7 9	71 67 39 52	32 15
36 6 18 73	18 -67	31 27 30 52	1 -25
2 16 13 19	-11 -3	6 62 43 60	-37 2
16 12 55 71	-39 -59	37 70 73 51	-36 19
56 43 69 43	-13 0	25 66 31 15	-6 51
2 53 6 24	-4 29	12 19 56 51	-44 -32
14 1 60 11	-46 -10	11 8 70 2	-59 6
57 17 32 37	25 -20	38 23 20 0	18 23
61 10 31 51	30 -41	7 19 52 3	-45 16
37 25 34 15	3 10	44 48 21 69	23 -21
13 3 5 73	8 -70	47 48 14 38	33 10
14 74 60 1	-46 73	51 34 5 47	46 -13
36 19 31 6	5 13	49 1 4 35	45 -34
6 32 24 39	-18 -7	50 74 14 43	36 31
38 30 55 66	-17 -36	56 50 39 51	17 -1
19 19 66 45	-47 -26	48 1 8 12	40 -11
12 2 57 52	-45 -50	71 41 60 11	11 30
2 34 64 38	-62 -4	69 55 29 72	40 -17
52 60 13 27	39 33	58 28 74 74	-16 -46
19 18 73 70	-54 -52	61 14 31 62	30 -48
67 37 55 49	12 -12	26 37 34 21	-8 16
66 60 18 40	48 20	22 7 52 63	-30 -56
11 25 73 60	-62 -35	17 41 5 52	12 -11
17 57 13 14	4 43	38 41 57 3	-19 38
1 47 40 48	-39 -1	35 6 2 13	33 -7
12 73 43 41	-31 32	20 11 62 10	-42 1
37 61 21 70	16 -9	66 29 52 1	14 28
15 23 13 20	2 3	5 64 23 6	-18 58
63 55 67 1	-4 54	41 31 71 68	-30 -37

0 47	24 44	-24 3	28 73 54 51 -26 22
47 47	71 12	-24 35	13 53 45 58 -32 -5
26 24	24 41	2 -17	27 45 30 31 -3 14
48 33	29 37	19 -4	11 3 2 60 9 -57
56 4	54 0	2 4	16 18 73 21 -57 -3
9 13	60 66	-51 -53	35 40 30 33 5 7
47 65	32 7	15 58	5 43 15 24 -10 19
26 55	3 49	23 6	1 72 45 33 -44 39
4 68	59 42	-55 26	67 35 56 66 11 -31
1 53	57 26	-56 27	8 32 23 18 -15 14
14 53	65 33	-51 20	46 8 2 30 44 -22
32 39	23 12	9 27	43 17 11 37 32 -20
14 22	41 30	-27 -8	25 71 40 49 -15 22
4 72	38 50	-34 22	29 55 70 7 -41 48
44 34	38 33	6 1	58 69 71 10 -13 59
4 15	2 24	2 -9	19 21 37 72 -18 -51
45 41	16 7	29 34	32 39 36 51 -4 -12
13 14	6 50	7 -36	53 30 58 12 -5 18
74 39	40 47	34 -8	40 42 26 51 14 -9
34 61	64 16	-30 45	28 51 59 22 -31 29
40 23	2 69	38 -46	68 26 10 35 58 -9
38 54	44 42	-6 12	10 43 15 11 -5 32
70 6	73 30	-3 -24	29 9 34 22 -5 -13
30 58	61 45	-31 13	8 73 10 14 -2 59
45 59	49 71	-4 -12	73 25 62 41 11 -16
58 67	67 24	-9 43	10 64 37 49 -27 15
4 45	22 14	-18 31	72 63 73 8 -1 55
68 4	66 47	2 -43	30 66 31 54 -1 12
53 62	57 11	-4 51	12 15 24 26 -12 -11
24 20	73 51	-49 -31	48 14 58 13 -10 1
32 51	70 61	-38 -10	55 8 15 54 40 -46
67 30	38 4	29 26	12 34 48 18 -36 16
74 10	73 28	1 -18	46 7 65 9 -19 -2
60 24	59 30	1 -6	54 3 2 41 52 -38
38 13	30 23	8 -10	51 53 34 34 17 19
73 34	49 31	24 3	4 8 10 12 -6 -4
21 70	9 59	12 11	43 69 37 71 6 -2
17 41	0 3	17 38	53 74 52 63 1 11
31 19	57 28	-26 -9	3 74 64 29 -61 45
45 37	58 74	-13 -37	69 54 41 27 28 27
72 56	18 52	54 4	58 25 60 46 -2 -21
18 19	10 50	8 -31	73 47 52 72 21 -25
0 63	62 63	-62 0	28 0 70 52 -42 -52
18 25	14 13	4 12	70 56 56 64 14 -8
34 16	51 71	-17 -55	35 12 51 28 -16 -16
49 2	45 27	4 -25	32 34 64 17 -32 17
69 44	51 52	18 -8	14 15 23 47 -9 -32
33 52	62 27	-29 25	41 39 28 40 13 -1
65 70	22 44	43 26	41 52 74 60 -33 -8
20 69	30 62	-10 7	22 61 36 10 -14 51
69 49	53 37	16 12	48 17 38 54 10 -37
17 62	61 27	-44 35	2 6 45 42 -43 -36
18 17	62 66	-44 -49	74 33 3 31 71 2
41 67	20 9	21 58	36 68 22 50 14 18
55 33	66 66	-11 -33	49 26 17 57 32 -31
45 20	22 6	23 14	67 43 22 15 45 28
21 63	27 7	-6 56	29 42 40 54 -11 -12
65 25	26 27	39 -2	54 72 36 32 18 40
13 48	39 48	-26 0	19 20 8 3 11 17
9 11	13 21	-4 -10	68 13 7 9 61 4

Table A.1: The first ten computer-generated random sets of 60 pixel-pairs
Included in each box is the distances in the 'x' and 'y'-directions between the pixels in each pair

Results tables for pixel-pairs experiments

In Table A.2, scores of less than 100% are highlighted in red. In subsequent tables, if the score is 100% and is an improvement on the corresponding score in Table A2, it is shown in blue. If the score is 100% and is the same as in Table A2, it is shown in black. Scores of less than 100% that are an improvement on the corresponding scores in Table A2 appear in green and scores that are lower are indicated in purple.

Set	60 random pairs Scores:			All four configurations Average Error: Circle			Diamond			Square			Number of pairs of each configuration: 0			1			2			3		
	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S
1	88/88	87/88	88/88	2	25	13	26	10	32	12	34	4	1	10	0	7	12	2	6	11	1	44	24	56
2	88/88	88/88	88/88	4	28	17	29	8	39	14	41	6	2	9	0	6	17	3	9	12	1	4	19	54
3	88/88	80/88	84/88	5	28	15	27	10	35	16	37	7	2	13	0	7	13	2	9	14	3	40	19	53
4	88/88	85/88	88/88	6	33	18	28	12	38	19	42	6	1	15	0	10	15	2	10	11	3	37	17	53
5	88/88	88/88	88/88	5	28	17	28	11	37	17	39	5	2	12	0	6	10	1	11	16	2	40	19	56
6	88/88	88/88	88/88	8	24	23	26	5	42	24	41	5	3	11	0	12	19	3	9	16	1	34	12	55
7	88/88	88/88	88/88	4	26	20	26	9	37	19	39	6	3	14	0	8	11	2	11	13	2	36	20	55
8	88/88	88/88	88/88	8	26	23	23	2	37	23	41	8	3	11	0	11	17	2	11	15	3	34	14	53
9	87/88	88/88	88/88	10	30	16	31	8	39	14	43	4	1	11	0	8	16	2	10	15	1	38	16	55
10	88/88	88/88	88/88	8	29	14	27	8	34	16	39	5	3	12	0	5	11	1	8	15	3	42	21	54

Table A.2: All 4 pixel-pair configurations and 60 random pairs

Set	60 random pairs Scores:			Configurations '3' and 'not 3' Average Error: Circle			Diamond			Square			Number of pairs of configuration 3:		
	C	D	S	C	D	S	C	D	S	C	D	S	Circles	Diamonds	Squares
1	88/88	88/88	88/88	2	20	11	21	5	29	11	32	4	44	24	56
2	88/88	88/88	88/88	4	24	15	26	5	36	13	39	5	41	19	54
3	88/88	87/88	88/88	4	22	14	22	4	32	16	35	7	40	19	53
4	88/88	88/88	88/88	5	22	17	21	4	35	18	40	6	37	17	53
5	88/88	88/88	88/88	5	21	17	22	7	34	17	37	5	40	19	56
6	88/88	88/88	88/88	6	20	23	21	5	42	24	41	5	34	12	55
7	88/88	88/88	88/88	4	17	18	18	4	33	19	37	6	36	20	55
8	88/88	88/88	88/88	6	20	21	20	0	34	22	39	8	34	14	53
9	88/88	88/88	88/88	9	24	16	27	5	38	13	42	4	38	16	55
10	88/88	88/88	88/88	6	23	14	22	4	31	15	36	5	42	21	54

Table A.3: Pixel-pair configurations '3' and 'not 3' and 60 random pairs

Set	100 random pairs Scores:			Configurations '3' and 'not 3' Average Error: Circle			Diamond			Square			Number of pairs of configuration 3:		
	C	D	S	C	D	S	C	D	S	C	D	S	Circles	Diamonds	Squares
1	88/88	88/88	87/88	6	35	19	36	9	51	20	57	7	72	37	92
2	88/88	88/88	88/88	9	34	31	39	9	60	24	62	9	66	34	91
3	88/88	88/88	88/88	11	34	27	38	8	54	27	61	10	63	30	88
4	88/88	88/88	88/88	11	35	26	35	8	55	29	63	10	61	29	88
5	88/88	88/88	88/88	7	35	25	37	11	52	25	61	9	67	34	91
6	88/88	88/88	88/88	10	35	34	37	10	64	36	68	11	58	22	90
7	88/88	88/88	88/88	10	34	28	37	6	57	27	64	10	62	31	91
8	88/88	88/88	88/88	9	34	32	33	4	57	33	64	10	62	29	92
9	88/88	88/88	87/88	16	36	27	42	7	58	23	66	9	60	27	90
10	88/88	88/88	88/88	11	34	27	37	8	53	25	57	9	69	37	90

Table A.4: Pixel-pair configurations '3' and 'not 3' and 100 random pairs

Set	100 random pairs Scores:			All four configurations Average Error: Circle			Diamond			Square			Number of pairs of each configuration: 0			1			2			3		
	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S
1	88/88	88/88	88/88	7	44	22	45	15	56	22	60	8	2	19	0	12	21	3	12	21	3	72	37	92
2	88/88	88/88	88/88	10	45	32	49	15	64	26	67	9	4	20	0	10	23	5	18	22	2	66	34	91
3	88/88	84/88	88/88	12	43	29	46	16	58	29	64	12	4	21	0	10	19	4	21	29	6	63	30	88
4	88/88	87/88	88/88	13	51	28	48	21	62	30	68	11	1	24	0	16	24	4	20	21	5	61	29	88
5	88/88	88/88	88/88	8	45	26	46	17	58	26	64	10	4	18	0	11	16	2	16	29	5	67	34	91
6	88/88	88/88	88/88	13	46	36	48	12	70	37	71	11	5	21	0	19	29	4	17	27	4	58	22	90
7	88/88	88/88	88/88	11	44	31	46	15	62	29	67	10	4	22	0	14	23	5	18	22	3	62	31	91
8	88/88	88/88	88/88	12	44	34	40	8	61	34	67	10	5	21	0	16	25	3	15	23	4	62	29	92
9	88/88	88/88	88/88	18	49	28	50	15	63	25	71	9	5	24	0	16	25	5	17	22	4	60	27	90
10	88/88	88/88	88/88	13	42	27	44	16	56	26	61	9	3	17	0	11	19	4	15	25	4	69	37	90

Table A.5: All 4 pixel-pair configurations and 100 random pairs

Set	60 random pairs Scores:			Configurations '1 and 2' and 'not 1 and 2' Average Error: Circle			Diamond			Square			Number of pairs of configurations 1 and 2 together:		
	C	D	S	C	D	S	C	D	S	C	D	S	Circles	Diamonds	Squares
1	88/88	88/88	88/88	2	20	11	21	10	24	11	24	4	13	23	3
2	88/88	88/88	88/88	4	24	15	25	8	31	13	32	5	15	29	4
3	88/88	81/88	79/88	5	20	14	22	9	25	14	22	7	16	27	5
4	88/88	79/88	87/88	6	29	18	21	11	26	17	28	6	20	26	5
5	88/88	87/88	88/88	5	26	15	23	10	27	15	29	5	17	26	3
6	88/88	88/88	88/88	7	21	19	22	4	30	22	31	5	21	35	4
7	88/88	86/88	88/88	4	23	19	21	8	25	18	26	6	19	24	4
8	88/88	88/88	88/88	8	23	21	19	2	29	19	30	7	22	32	5
9	85/88	88/88	88/88	10	23	15	25	8	28	13	29	4	18	31	3
10	80/88	88/88	88/88	8	24	10	23	7	24	13	27	15	13	26	4

Table A.6: Pixel-pair configurations '1 and 2' and 'not 1 and 2' and 60 random pairs

Set	100 random pairs Scores:			Configurations '1 and 2' and 'not 1 and 2' Average Error: Circle			Diamond			Square			Number of pairs of configurations 1 and 2 together:		
	C	D	S	C	D	S	C	D	S	C	D	S	Circles	Diamonds	Squares
1	88/88	88/88	88/88	6	33	19	36	14	39	20	39	7	24	42	6
2	88/88	88/88	88/88	10	41	28	42	14	47	23	50	8	28	45	7
3	88/88	83/88	88/88	12	30	27	38	15	42	26	43	11	31	48	10
4	88/88	79/88	87/88	8	44	23	41	16	45	22	52	9	27	45	7
5	88/88	87/88	88/88	13	41	31	42	11	54	34	54	10	36	56	8
6	88/88	88/88	88/88	11	38	29	38	14	45	26	50	10	32	45	8
7	88/88	88/88	88/88	12	35	30	32	8	45	29	46	10	31	48	7
8	88/88	88/88	88/88	18	39	26	38	14	45	22	48	9	33	47	9
9	88/88	88/88	88/88	12	36	23	37	14	42	23	46	9	26	44	8
10	88/88	88/88	88/88	12	43	26	38	21	43	28	46	11	36	45	9

Table A.7: Pixel-pair configurations '1 and 2' and 'not 1 and 2' and 100 random pairs

Set	130 random pairs Scores:			Configurations '1 and 2' and 'not 1 and 2' Average Error: Circle			Diamond			Square			Number of pairs of configurations 1 and 2 together:		
	C	D	S	C	D	S	C	D	S	C	D	S	Circles	Diamonds	Squares
1	88/88	88/88	88/88	9	44	26	47	18	51	25	52	10	31	54	8
2	88/88	88/88	88/88	13	55	35	57	17	62	35	62	12	39	59	10
3	88/88	88/88	88/88	17	42	38	50	19	57	37	59	14	41	62	13
4	88/88	72/88	88/88	17	53	33	46	27	56	37	59	13	45	59	12
5	88/88	88/88	88/88	11	55	32	52	19	60	32	68	14	40	67	12
6	88/88	88/88	88/88	15	48	42	48	14	63	44	63	13	45	63	11
7	88/88	88/88	88/88	14	47	37	47	19	58	34	62	11	40	56	9
8	88/88	88/88	88/88	14	44	37	43	12	60	36	60	13	36	60	9
9	88/88	88/88	88/88	22	51	34	50	18	54	31	59	10	42	58	10
10	88/88	88/88	88/88	16	48	29	49	19	60	29	64	9	32	63	10

Table A.8: Pixel-pair configurations '1 and 2' and 'not 1 and 2' and 130 random pairs

Set	60 random pairs Scores:			All four configurations Average Error: Circle			Diamond			Square			Pixels <= 10 apart in both x and y-directions Number of pairs of each configuration: 0			1			2			3		
	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S
1	88/88	88/88	85/88	5	22	8	20	7	25	9	25	4	6	20	1	1	3	0	4	4	2	47	31	55
2	88/88	78/88	88/88	5	26	10	23	9	26	10	31	4	6	19	1	2	6	1	2	3	0	49	30	56
3	88/88	71/88	82/88	2	23	9	21	11	25	7	27	3	4	17	0	2	5	2	1	3	0	52	33	57
4	80/88	88/88	88/88	5	25	8	26	8	29	7	28	2	3	16	0	3	6	1	1	7	0	52	29	57
5	88/88	88/88	88/88	4	21	10	22	7	26	9	29	5	7	19	0	2	5	1	0	3	1	48	31	56
6	88/88	88/88	87/88	4	24	14	26	8	27	11	27	4	2	20	0	2	5	1	8	3	2	46	30	55
7	87/88	87/88	88/88	4	20	8	19	10	24	8	26	2	4	15	0	4	5	1	0	3	0	50	35	57
8	88/88	88/88	88/88	5	28	16	27	10	34	16	36	5	8	25	0	4	4	2	3	5	0	44	24	55
9	88/88	88/88	88/88	6	26	14	23	8	30	12	34	4	7	21	0	5	5	2	2	4	1	44	28	56
10	88/88	88/88	88/88	1	23	9	22	8	28	9	28	1	5	17	0	0	6	0	3	4	1	50	31	58

Table A.9: All four pixel-pair configurations, pixels <= 10 apart and 60 random pairs

Set	60 random pairs Scores:			All four configurations Average Error: Circle			Diamond			Square			Pixels <= 5 apart in both x and y-directions Number of pairs of each configuration: 0			1			2			3		
	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S
1	88/88	86/88	88/88	5	20	10	21	7	27	10	28	2	9	23	1	1	2	0	1	2	0	47	30	57
2	88/88	73/88	88/88	5	21	10	20	10	25	7	27	1	5	18	0	1	4	0	1	2	0	50	34	58
3	88/88	88/88	83/88	7	22	16	24	8	31	14	32	6	10	26	0	1	2	2	1	2	1	46	28	55
4	88/88	88/88	88/88	7	21	13	20	8	27	12	29	3	8	23	0	1	2	1	3	1	0	46	32	56
5	69/88	86/88	88/88	4	24	7	23	11	25	6	26	2	2	18	0	1	5	0	2	1	0	53	34	58
6	88/88	88/88	88/88	5	21	13	22	7	27	11	27	3	7	22	0	1	1	1	2	2	1	48	32	56
7	70/88	88/88	88/88	3	16	6	16	6	20	8	20	1	3	14	0	2	2	0	0	2	0	53	39	58
8	88/88	88/88	88/88	3	20	7	20	9	23	5	24	1	4	16	0	1	3	0	0	3	0	53	35	58
9	88/88	80/88	88/88	6	20	12	17	9	27	12	28	3	8	21	0	2	2	1	2	3	0	46	32	56
10	88/88	88/88	88/88	5	21	10	20	7	24	7	26	1	4	20	0	1	1	0	3	2	1	49	34	58

Table A.10: All four pixel-pair configurations, pixels <= 5 apart and 60 random pairs

Set	60 random pairs Scores:			All four configurations Average Error: Circle			Diamond			Square			Pixels <= 3 apart in both x and y-directions Number of pairs of each configuration: 0			1			2			3		
	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S
1	88/88	88/88	86/88	3	13	8	13	6	20	12	22	5	9	20	0	2	1	1	0	2	0	47	36	57
2	40/88	88/88	88/88	5	28	5	26	11	29	6	31	1	2	23	0	2	2	0	1	2	0	53	30	58
3	85/88	88/88	87/88	8	26	13	28	8	31	13	33	6	8	30	1	2	2	1	1	1	1	47	24	55
4	88/88	87/88	88/88	5	17	10	18	6	24	9	25	1	7	22	0	2	1	1	0	0	0	49	34	58
5	76/88	87/88	86/88	5	23	8	21	8	25	9	28	4	4	22	1	2	1	1	0	2	0	52	33	57
6	88/88	85/88	88/88	7	23	12	21	9	26	11	31	3	9	23	1	1	2	1	1	2	1	48	30	56
7	88/88	88/88	88/88	5	17	12	18	6	25	13	27	5	10	24	1	2	1	0	0	1	0	46	32	57
8	88/88	88/88	88/88	3	16	7	15	7	20	7	22	1	6	18	0	2	1	0	0	1	0	51	38	58
9	88/88	88/88	88/88	2	19	8	18	7	24	8	24	3	5	20	0	2	1	1	0	2	0	50	35	57
10	51/88	88/88	88/88	5	23	7	22	7	24	8	26	0	3	21	0	1	1	0	1	1	0	53	34	59

Table A.11: All four pixel-pair configurations, pixels <= 3 apart and 60 random pairs

	150 random pairs			All four configurations									Pixels <= 10 apart in both x and y-directions											
	Scores:			Average Error: Circle			Diamond			Square			Number of pairs of each configuration: 0			1			2			3		
	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S
1	88 88	88 88	88 88	11	58	20	61	23	69	20	72	7	14	45	1	3	14	2	7	13	4	125	76	141
2	88 88	88 88	88 88	12	58	20	59	21	62	18	69	8	11	40	1	5	16	2	6	9	3	127	83	142
3	88 88	88 88	88 88	13	56	29	56	22	69	27	71	11	13	30	1	7	10	4	5	8	3	122	80	140
4	88 88	88 88	88 88	14	67	31	69	21	77	25	79	9	12	49	0	8	13	2	9	16	4	120	70	141
5	88 88	87 88	88 88	10	54	31	53	21	66	30	69	11	15	46	1	8	12	4	5	9	3	120	81	140
6	88 88	88 88	88 88	12	56	28	59	19	62	25	63	9	7	45	1	10	12	4	9	9	2	122	82	141
7	88 88	88 88	88 88	11	56	27	55	25	66	28	71	8	11	42	1	11	14	3	4	11	3	122	82	142
8	88 88	88 88	88 88	16	72	34	72	27	83	32	88	12	15	57	0	11	10	4	9	17	3	113	64	140
9	88 88	88 88	88 88	12	64	28	59	23	71	24	78	8	14	46	0	10	11	3	4	15	3	120	76	142
0	88 88	88 88	88 88	10	56	24	56	22	70	24	70	7	14	47	0	4	11	2	7	12	3	123	78	142

Table A.12: All four pixel-pair configurations, pixels <= 10 apart and 150 random pairs

	150 random pairs			All four configurations									No restriction on distance between pixels											
	Scores:			Average Error: Circle			Diamond			Square			Number of pairs of each configuration: 0			1			2			3		
	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S
1	88 88	88 88	88 88	10	61	39	60	23	81	36	88	12	6	27	0	21	33	7	18	29	3	104	59	138
2	88 88	88 88	88 88	15	68	48	69	21	94	44	100	15	7	31	0	18	33	7	28	35	4	94	49	137
3	88 88	88 88	88 88	19	67	47	70	24	91	46	101	18	6	31	0	20	32	6	27	42	8	95	44	133
4	88 88	88 88	88 88	19	74	41	67	29	91	46	101	16	2	33	0	23	32	7	29	38	7	93	44	134
5	88 88	88 88	88 88	13	69	42	67	25	87	41	97	15	5	26	0	15	28	4	29	46	9	98	48	135
6	88 88	88 88	88 88	18	68	54	70	18	102	55	103	15	9	38	1	24	33	5	27	38	7	88	38	135
7	88 88	88 88	88 88	15	67	43	66	25	90	43	98	13	6	31	0	23	36	8	23	29	3	96	52	137
8	88 88	88 88	88 88	18	63	47	62	13	91	47	98	17	7	30	0	22	35	5	23	37	6	96	47	138
9	88 88	88 88	88 88	24	73	44	75	21	95	40	105	13	6	34	0	25	35	6	22	36	5	95	43	137
0	88 88	88 88	88 88	18	66	40	68	26	89	40	94	11	3	24	0	22	40	6	19	32	5	104	52	137

Table A.13: All four pixel-pair configurations, no distance restriction and 150 random pairs

	150 random pairs			All four configurations									No restriction on distance between pixels											
	Scores:			Average Error: Circle			Diamond			Square			Number of pairs of each configuration: 0			1			2			3		
	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S	C	D	S
1	88 88	88 88	88 88	15	67	49	69	26	94	48	102	16	5	32	0	24	39	4	25	34	7	94	43	137
2	88 88	88 88	88 88	19	67	47	70	24	91	46	101	18	6	31	0	20	32	6	27	42	8	95	44	133
3	88 88	88 88	88 88	13	63	39	63	22	85	35	90	11	5	27	0	16	29	3	24	35	6	103	57	140
4	88 88	88 88	88 88	20	60	48	68	16	97	44	101	14	4	24	0	25	36	5	26	47	3	93	42	140
5	88 88	88 88	88 88	19	63	45	70	22	89	39	94	16	5	28	0	20	32	7	22	38	6	101	51	136
6	88 88	88 88	88 88	18	80	45	73	27	97	46	109	18	6	36	0	22	37	6	24	36	8	96	39	134
7	88 88	88 88	88 88	19	69	48	69	19	92	48	101	16	5	31	0	20	29	5	30	43	6	94	44	138
8	88 88	88 88	88 88	12	65	48	64	21	94	50	98	17	5	30	0	20	36	6	26	36	6	97	45	136
9	88 88	88 88	88 88	17	73	46	75	28	98	45	103	17	3	34	0	22	33	7	25	40	5	97	40	136
0	88 88	88 88	88 88	19	63	58	70	23	98	47	97	16	8	36	0	25	38	9	24	29	5	91	45	135

Table A.14: All four pixel-pair configurations, no distance restriction, 2nd set of 150 random pairs

Appendix B: Exploring the multi-level architecture - results

Squares test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Training Object matched
Object 0	8	8	8	**100**	0
Object 1	8	8	8	**100**	0
Object 2	6	6	6	**100**	2
Object 3	2	2	2	**100**	25
Object 4	2	2	2	**100**	11
Object 5	4	4	4	**100**	1
Object 6	6	6	6	**100**	17
Object 7	7	6	4	**57**	17
Object 8	6	6	6	**100**	8
Object 9	6	6	6	**100**	2
Object 10	6	6	6	**100**	13
Object 11	4	4	4	**100**	20
Object 12	4	4	4	**100**	1
Object 13	4	4	4	**100**	20
Object 14	4	4	4	**100**	5
Object 15	2	4	1	**25 !**	31
Object 16	4	4	4	**100**	1
Object 17	4	4	4	**100**	20
Object 18	4	4	4	**100**	6
Object 19	6	6	6	**100**	8
Object 20	5	5	5	**100**	10
Object 21	4	4	4	**100**	6
Object 22	4	4	4	**100**	6
Object 23	6	6	6	**100**	13
Object 24	4	4	2	**50**	1
Object 25	4	4	4	**100**	20
Object 26	5	5	5	**100**	10
Object 27	5	5	5	**100**	10
Object 28	4	4	4	**100**	20
Object 29	4	4	4	**100**	20
Object 30	4	4	3	**75**	6
Object 31	4	4	4	**100**	6
Object 32	4	4	4	**100**	6
Object 33	5	5	5	**100**	10
Object 34	6	6	6	**100**	13
Object 35	6	6	6	**100**	19
Object 36	6	6	6	**100**	9
Object 37	6	6	6	**100**	7
Object 38	6	6	6	**100**	2
Object 39	8	8	8	**100**	0
Object 40	8	8	8	**100**	0
Object 41	8	8	8	**100**	0
Object 42	8	8	8	**100**	0
Object 43	6	6	6	**100**	16
Object 44	6	6	5	**83**	17
Object 45	6	6	6	**100**	9
Object 46	8	8	8	**100**	0
Object 47	6	6	6	**100**	19
Object 48	6	6	6	**100**	18
Object 49	8	8	8	**100**	0

Table B.1(a): Test squares results for the 'whole object' matching scheme

The matched training objects that are numbered '25' or less in the rightmost column are squares. Those numbered higher than '25' are circles. Forty-five of the fifty squares are matched 100% correctly. Square 15 is misclassified as a circle but with a score of just 25% - marked '!'.

Circles test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Training Object matched
Object 0	8	8	8	**100**	62
Object 1	4	4	4	**100**	65
Object 2	6	6	6	**100**	37
Object 3	2	4	1	**25**	55
Object 4	4	4	4	**100**	49
Object 5	4	4	4	**100**	50
Object 6	4	4	4	**100**	27
Object 7	4	4	4	**100**	51
Object 8	4	4	4	**100**	50
Object 9	4	4	4	**100**	32
Object 10	4	4	4	**100**	44
Object 11	2	2	2	**100**	41
Object 12	3	3	3	**100**	52
Object 13	4	4	4	**100**	50
Object 14	2	2	2	**100**	41
Object 15	4	4	4	**100**	40
Object 16	4	4	4	**100**	50
Object 17	4	4	4	**100**	51
Object 18	2	2	2	**100**	41
Object 19	4	4	4	**100**	42
Object 20	4	4	4	**100**	28
Object 21	3	3	3	**100**	52
Object 22	4	4	4	**100**	29
Object 23	2	2	2	**100**	41
Object 24	4	4	4	**100**	47
Object 25	4	4	4	**100**	51
Object 26	4	4	4	**100**	32
Object 27	4	4	4	**100**	32
Object 28	6	6	6	**100**	37
Object 29	8	8	8	**100**	45
Object 30	6	6	6	**100**	35
Object 31	4	4	4	**100**	54
Object 32	4	4	4	**100**	55
Object 33	4	4	4	**100**	56
Object 34	4	4	4	**100**	57
Object 35	4	4	4	**100**	32
Object 36	2	4	1	**25**	31
Object 37	4	4	4	**100**	44
Object 38	4	4	4	**100**	51
Object 39	4	4	4	**100**	51
Object 40	6	6	6	**100**	37
Object 41	8	8	8	**100**	61
Object 42	6	6	6	**100**	60
Object 43	3	3	3	**100**	52
Object 44	4	4	4	**100**	55
Object 45	4	4	4	**100**	28
Object 46	2	-	0	**0**	-
Object 47	3	2	1	**33**	43
Object 48	4	4	2	**50**	57
Object 49	2	3	1	**33**	52

Table B.1(b): Test circles results for the 'whole object' matching scheme

Forty-four out of the fifty circles are matched 100% correctly. The remaining six scores are 50% or less, including one 0% recognition. There are no misclassifications.

Polygons and ellipses test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	Percent score	Training Object matched
Object 0	7	8	4	**50 !**	62
Object 1	2	-	0	**0**	-
Object 2	2	6	1	**16**	16
Object 3	7	8	3	**37**	45
Object 4	6	8	3	**37**	62
Object 5	4	6	1	**16**	13
Object 6	4	4	1	**25**	27
Object 7	6	4	1	**16**	24
Object 8	4	4	1	**25**	1
Object 9	8	8	2	**25**	0
Object 10	6	4	1	**16**	27
Object 11	7	4	2	**28**	47
Object 12	8	8	2	**25**	0
Object 13	6	6	1	**16**	3
Object 14	2	-	0	**0**	-
Object 15	7	6	3	**42**	2
Object 16	4	6	2	**33**	13
Object 17	2	4	1	**25**	29
Object 18	6	4	2	**33**	28
Object 19	6	6	3	**50 !**	13
Object 20	4	-	0	**0**	-
Object 21	2	-	0	**0**	-
Object 22	4	4	1	**25**	29
Object 23	2	4	1	**25**	5
Object 24	5	5	3	**60 !**	10

Table B.1(c): Polygons and ellipses results for the 'whole object' matching scheme

There are no matches above 60%. The three matches of 50% or greater are marked '!'.

Squares test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Object matched	Hybrid match - overall construct match %	Hybrid match - number of square matches	Hybrid match – number of circle matches	**Hybrid Percent Square score**	**Hybrid Percent Circle score**
Square 7	7	6	4	**57**	17 (S)	85	25	23	**44**	-
Square 15	2	4	1	**25 !**	31 (C)	50	0	4	-	**50 !**
Square 24	4	4	2	**50**	1 (S)	75	8	15	-	**48 !**
Square 30	4	4	3	**75**	6 (S)	100	30	4	**88 ***	-
Square 44	6	6	5	**83**	17 (S)	100	49	19	**72 ^**	-

Table B.2(a): Comparing the scores of the five test squares not completely matched under the 'whole object' scheme with their scores with 'hybrid matching 1'

'S' = square, 'C' = circle. Scores raised above the 80% threshold by the hybrid scheme are marked '*', scores taken below the threshold by the hybrid scheme are marked '^' and misclassifications are indicated by '!'

Circles test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Object matched	Hybrid match - overall construct match %	Hybrid match - number of square matches	Hybrid match – number of circle matches	**Hybrid Percent Square score**	**Hybrid Percent Circle score**
Circle 3	2	4	1	**25**	55 (C)	100	1	3	-	75
Circle 36	2	4	1	**25**	31 (C)	50	0	4	-	50
Circle 47	3	2	1	**33**	43 (C)	66	0	7	-	66
Circle 48	4	4	2	**50**	57 (C)	100	1	19		95 *
Circle 49	2	3	1	**33**	52 (C)	50	0	13	-	50

Table B.2(b): Comparing the scores of the five non-zero scoring test circles that were not completely matched under the 'whole object' scheme with their scores with 'hybrid matching 1'

Scores raised above the 80% threshold by the hybrid scheme are marked '*'.

Polygons and Ellipses test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Object matched	Hybrid match - overall construct match %	Hybrid match - number of square matches	Hybrid match – number of circle matches	**Hybrid Percent Square score**	**Hybrid Percent Circle score**
Object 0	7	8	4	**50**	62 (C)	85	3	12	-	68 ↑
Object 1	2	-	0	**0**	-	0	0	0	0	0
Object 2	2	6	1	**16**	16 (S)	50	3	0	50 ↑	-
Object 3	7	8	3	**37**	45 (C)	85	12	31	-	61 ↑
Object 4	6	8	3	**37**	62 (C)	66	16	9	42 ↑	-
Object 5	4	6	1	**16**	13 (S)	50	3	0	50 ↑	-
Object 6	4	4	1	**25**	27 (C)	25	0	15	-	25
Object 7	6	4	1	**16**	24 (S)	50	3	6	-	33 ↑
Object 8	4	4	1	**25**	1 (S)	75	7	0	75 ↑	-
Object 9	8	8	2	**25**	0 (S)	75	25	13	47 ↑	-
Object 10	6	4	1	**16**	27 (C)	50	2	9	-	40 ↑
Object 11	7	4	2	**28**	47 (C)	71	8	9	-	37 ↑
Object 12	8	8	2	**25**	0 (S)	50	21	7	37 ↑	-
Object 13	6	6	1	**16**	3 (S)	16	2	0	16	-
Object 14	2	-	0	**0**	-	0	0	0	0	0
Object 15	7	6	3	**42**	2 (S)	100	33	27	55 ↑	-
Object 16	4	6	2	**33**	13 (S)	50	5	0	50 ↑	-
Object 17	2	4	1	**25**	29 (C)	50	10	6	31 ↑	-
Object 18	6	4	2	**33**	28 (C)	50	4	8	-	33
Object 19	6	6	3	**50**	13 (S)	83	17	15	44 ↓	-
Object 20	4	-	0	**0**	-	0	0	0	0	0
Object 21	2	-	0	**0**	-	0	0	0	0	0
Object 22	4	4	1	**25**	29 (C)	25	0	3	-	25
Object 23	2	4	1	**25**	5 (S)	50	3	0	50 ↑	-
Object 24	5	5	3	**60**	10 (S)	100	34	14	70 ↑	-

Table B.2(c): Comparing the 'whole object' scores of the polygons and ellipses with their scores under 'hybrid matching 1'.

The highlighted columns show that several of the scores are increased with the hybrid scheme, marked '↑'. Lowered hybrid scores are marked '↓'.

Squares test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Object matched	Hybrid match - number of square constructs matched	Hybrid match – number of circle constructs matched	Hybrid match – number of 'mixed' constructs matched	**Hybrid Percent Square score**	**Hybrid Percent Circle score**
Square 7	7	6	4	57	17 (S)	5	0	1	77	-
Square 15	2	4	1	25.1	31 (C)	0	1	0	-	50.1
Square 24	4	4	2	50	1 (S)	2	0	1	62	-
Square 30	4	4	3	75	6 (S)	3	0	1	87 *	-
Square 44	6	6	5	83 *	17 (S)	5	0	1	91 *	-

Table B.3(a): Comparing the scores of the five test squares not completely matched under the 'whole object' scheme with their scores with 'hybrid matching 2'

All the scores are raised under this hybrid scheme, with two of them being raised above the 80% threshold – marked '*'. Also there is just the one hybrid misclassification - marked '!' – compared with two under 'hybrid matching 1' in Table B.2(a).

Circles test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Object matched	Hybrid match - number of square constructs matched	Hybrid match – number of circle constructs matched	Hybrid match – number of 'mixed' constructs matched	**Hybrid Percent Square score**	**Hybrid Percent Circle score**
Circle 3	2	4	1	25	55 (C)	0	1	1	-	100 *
Circle 36	2	4	1	25	51 (C)	0	1	0	-	50
Circle 47	3	2	1	33	45 (C)	0	2	0	-	66
Circle 48	4	4	2	50	57 (C)	0	3	1		100 *
Circle 49	2	3	1	33	52 (C)	0	1	0	-	50

Table B.3(b): Comparing the scores of the five non-zero scoring test circles not completely matched under the 'whole object' scheme with their scores with 'hybrid matching 2'

All the scores are raised under this hybrid scheme, with two of them being raised to 100% – marked '*'.

Polygons and Ellipses test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Object matched	Hybrid match - number of square constructs matched	Hybrid match – number of circle constructs matched	Hybrid match – number of 'mixed' constructs matched	**Hybrid Percent Square score**	**Hybrid Percent Circle score**
Object 0	7	8	4	50	62 (C)	1	5	0	-	70 ↑
Object 1	2	-	0	0	-	0	0	0	0	0
Object 2	2	6	1	16	16 (S)	1	0	0	50 ↑	-
Object 3	7	8	3	37	43 (C)	2	3	1	-	49 ↑
Object 4	6	8	3	37	62 (C)	0	4	0	-	66 ↑
Object 5	4	6	1	16	13 (S)	2	0	0	50 ↑	-
Object 6	4	4	1	25	27 (C)	0	1	0	-	25
Object 7	6	4	1	16	24 (S)	2	1	0	33 ↑	-
Object 8	4	4	1	25	1 (S)	3	0	0	75 ↑	-
Object 9	8	8	2	25	0 (S)	2	3	1	-	43 ↑
Object 10	6	4	1	16	27 (C)	1	2	0	-	33 ↑
Object 11	7	4	2	28	47 (C)	3	2	0	42 ↑	-
Object 12	8	8	2	25	0 (S)	2	0	2	31 ↑	-
Object 13	6	6	1	16	3 (S)	1	0	0	16	-
Object 14	2	-	0	0	-	0	0	0	0	0
Object 15	7	6	3	42	2 (S)	3	4	0	-	57 ↑
Object 16	4	6	2	33	13 (S)	2	0	0	50 ↑	-
Object 17	2	4	1	25	29 (C)	0	0	1	25	25
Object 18	6	4	2	33	28 (C)	1	1	1	25 ↓	25 ↓
Object 19	6	6	3	50	13 (S)	4	1	0	66 ↑	-
Object 20	4	-	0	0	-	0	0	0	0	0
Object 21	2	-	0	0	-	0	0	0	0	0
Object 22	4	4	1	25	29 (C)	0	1	0	-	25
Object 23	2	4	1	25	5 (S)	1	0	0	50 ↑	-
Object 24	5	5	3	60	10 (S)	0	3	2	-	80 ↑!

Table B.3(c): Comparing the 'whole object' scores of the polygons and ellipses with their scores under 'hybrid matching 2'

The highlighted columns show that most of the scores are increased with this hybrid scheme - marked '↑' – including the score for Object 24 which reaches the recognition threshold – marked '!'. Lowered hybrid scores are marked '↓'.

Squares test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Object matched	Hybrid match - number of square constructs matched	Hybrid match – number of circle constructs matched	Hybrid match – number of 'mixed' constructs matched	**Hybrid Percent Square score**	**Hybrid Percent Circle score**
Square 7	7	6	4	57	17 (S)	5	0	1	70 ↓	-
Square 15	2	4	1	25	31 (C)	0	1	0	-	50
Square 24	4	4	2	50	1 (S)	2	0	1	62	-
Square 30	4	4	3	75	6 (S)	3	0	1	87*	-
Square 44	6	6	5	83	17 (S)	5	0	1	100 ↑*	-

Table B.4(a): The scores of the five test squares not completely matched under the 'whole object' scheme and their scores with 'hybrid matching 3'

This time, the symbols in the 'Hybrid Percent Square score' column reflect comparison of 'hybrid matching 3' with 'hybrid matching 2' in Table B.3(a). The symbol '↑' indicates a raised score, '↓', a lowered score and '*' a score maintained above the recognition threshold.

Circles test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	**Percent score**	Object matched	Hybrid match - number of square constructs matched	Hybrid match – number of circle constructs matched	Hybrid match – number of 'mixed' constructs matched	**Hybrid Percent Square score**	**Hybrid Percent Circle score**
Circle 3	2	4	1	25	55 (C)	0	1	1	-	75 ↓
Circle 36	2	4	1	25	31 (C)	0	1	0	-	50
Circle 47	3	3	1	33	43 (C)	0	2	0	-	66
Circle 48	4	4	2	50	57 (C)	0	3	1		100 *
Circle 49	2	3	1	33	52 (C)	0	1	0	-	50

Table B.4(b): The scores of the five non-zero scoring test circles that were not completely matched under the whole object scheme

and their scores with 'hybrid matching 3'. This time, the symbols in the 'Hybrid Percent Circle score' column reflect comparison of 'hybrid matching 3' with 'hybrid matching 2' in Table 3(b). The symbol '↓' indicates a lowered score and '*', a score maintained above the recognition threshold.

Polygons and Ellipses test file	Number of Test Constructs	Number of Training Constructs	Matched Construct score	Percent score	Object matched	Hybrid match - number of square constructs matched	Hybrid match – number of circle constructs matched	Hybrid match – number of 'mixed' constructs matched	**Hybrid Percent Square score**	**Hybrid Percent Circle score**
Object 0	7	8	4	50	62 (C)	1	5	0	-	70
Object 1	2	-	0	0	-	0	0	0	0	0
Object 2	2	6	1	16	16 (S)	1	0	0	50	-
Object 3	7	8	3	37	45 (C)	2	3	1	-	49
Object 4	6	8	3	37	62 (C)	0	4	0	-	57 ↓
Object 5	4	6	1	16	13 (S)	2	0	0	50	-
Object 6	4	4	1	25	27 (C)	0	1	0	-	25
Object 7	6	4	1	16	24 (S)	2	1	0	25 ↓	-
Object 8	4	4	1	25	1 (S)	3	0	0	75	-
Object 9	8	8	2	25	0 (S)	2	3	1	-	43
Object 10	6	4	1	16	27 (C)	1	2	0	-	33
Object 11	7	4	2	28	47 (C)	3	2	0	35 ↓	-
Object 12	8	8	2	25	0 (S)	2	0	2	31	-
Object 13	6	6	1	16	5 (S)	1	0	0	16	-
Object 14	2	-	0	0	-	0	0	0	0	0
Object 15	7	6	3	42	2 (S)	3	4	0	-	57
Object 16	4	6	2	33	15 (S)	2	0	0	50	-
Object 17	2	4	1	25	29 (C)	0	0	1	25	25
Object 18	6	4	2	33	28 (C)	1	1	1	25	25
Object 19	6	6	3	50	15 (S)	4	1	0	66	-
Object 20	4	-	0	0	-	0	0	0	0	0
Object 21	2	-	0	0	-	0	0	0	0	0
Object 22	4	4	1	25	29 (C)	0	1	0	-	25
Object 23	2	4	1	25	5 (S)	1	0	0	50	-
Object 24	5	5	3	60	10 (S)	0	3	2	-	70 ↓*

Table B.4(c): The 'whole object matching' scores of the test polygons and ellipses

and their scores with 'hybrid matching 3'. This time, the symbols in the two rightmost columns of the table reflect comparison of 'hybrid matching 3' with 'hybrid matching 2' in Table B3(c). The symbol '↓' indicates a lowered score and '*' shows that the score has been taken back below the recognition threshold.

Recognition scheme	Squares			Circles			Polygons and Ellipses	
	% Recognised	% Rejected	% misclassified	% Recognised	% Rejected	% misclassified	% Rejected	% misclassified
Whole object	92	8	0	88	12	0	100	0
Hybrid 1	92	8	0	90	10	0	100	0
Hybrid 2	94	6	0	92	8	0	96	4
Hybrid 3	94	6	0	90	10	0	100	0

Table B.5: Overall performance of the four recognition schemes across the three test sets

assuming a recognition threshold of 80%

Appendix C Above-average scoring polygons for classification

(0s and 1s – results)

	0s			1s		
Best Win	Correct	Wrong	Non	Correct	Wrong	Non
**1166 (57 pixels)	1990/2000 = 99.5%	10/2000 = 0.5%	0	1996/2000 = 99.8%	4/2000 = 0.2%	0
116 5 pixels	1878/2000 = 93.9%	13/2000 = 0.65%	109/2000 = 5.45%	8/2000 = 0.4%	27/2000 = 1.35	1965/2000 = 98.25%
*1310 22 pixels	1162/2000 = 58.1%	15/2000 = 0.7%	824/2000	1977/2000 = 98.85%	23/2000 = 1.15%	0
207 21 pixels	1976/2000 = 98.8%	12/2000 = 0.6%	12/2000 = 0.6%	55/2000 = 2.75%	9/2000 = 0.45%	1936/2000 = 96.8%
**272 70 pixels	1989/2000 = 99.45%	11/2000 = 0.55%	0	1997/2000 = 99.85%	3/2000 = 0.15%	0
141 4 pixels	1963/2000 = 98.15%	6/2000 = 0.3%	31/2000 = 1.55%	0	11/2000 = 0.55%	1989/2000 = 99.45%
**1376 30 pixels	1976/2000 = 98.8%	24/2000 = 1.2%	0	1977/2000 = 98.85%	23/2000 = 1.15%	0
**52 24 pixels	1980/2000 = 99.0%	20/2000 = 1.0%	0	1985/2000 = 99.15%	16/2000 = 0.8%	1/2000 = 0.05%
**1270 26 pixels	1981/2000 = 99.05%	17/2000 = 0.85%	2/2000 = 0.1%	1967/2000 = 98.35%	26/2000 = 1.3%	7/2000 = 0.35%
729 14 pixels	1791/2000 = 89.55%	6/2000 = 0.3%	203/2000 = 10.15%	6/2000 = 0.3%	12/2000 = 0.6%	1982/2000 = 99.1%
227 21 pixels	1968/2000 = 98.4%	19/2000 = 0.95%	13/2000 = 0.65%	526/2000 = 26.3%	14/2000 = 0.7%	1460/2000 = 73%
183 5 pixels	1751/2000 = 87.55%	5/2000 = 0.25%	244/2000 = 12.2%	7/2000 = 0.35%	11/2000 = 0.55%	1982/2000 = 99.1%
*274 8 pixels	1592/2000 = 79.6%	86/2000 = 4.3%	322/2000 = 16.1%	1949/2000 = 97.45%	47/2000 = 2.35%	4/2000 = 0.2%
*1248 6 pixels	1205/2000 = 60.25%	27/2000 = 1.35%	768/2000 = 38.4%	1962/2000 = 98.1%	24/2000 = 1.2%	14/2000 = 0.7%
**1019 25 pixels	1977/2000 = 98.85%	21/2000 = 1.05%	2/2000 = 0.1%	1948/2000 = 97.4%	42/2000 = 2.1%	10/2000 = 0.5%
**80 27 pixels	1985/2000 = 99.25%	15/2000 = 0.75%	0	1989/2000 = 99.45%	11/2000 = 0.55%	0
**98 47 pixels	1978/2000 = 98.9%	22/2000 = 1.1%	0	1987/2000 = 99.35%	13/2000 = 0.65%	0
655 34 pixels	1977/2000 = 98.85%	19/2000 = 0.95%	4/2000 = 0.2%	485/2000 = 24.25%	7/2000 = 0.35%	1508/2000 = 75.4%
360 6 pixels	1893/2000 = 94.65%	13/2000 = 0.65%	94/2000 = 4.7%	85/2000 = 4.25%	32/2000 = 1.6%	1883/2000 = 94.15%
657 25 pixels	1964/2000 = 98.2%	24/2000 = 1.2%	12/2000 = 0.6%	419/2000 = 20.95%	16/2000 = 0.8%	1565/2000 = 78.25%
*187 13 pixels	1015/2000 = 50.75%	38/2000 = 1.9%	947/2000 = 47.35%	1971/2000 = 98.55%	24/2000 = 1.2%	5/2000 = 0.25%

*1536 9 pixels	1765/2000 = 88.15%	70/2000 = 3.5%	167/2000 = 8.35%	1915/2000 = 95.75%	51/2000 = 2.55%	34/2000 = 1.7%
*1136 17 pixels	1835/2000 = 91.75%	45/2000 = 2.15%	122/2000 = 6.1%	1379/2000 = 68.95%	27/2000 = 1.35%	594/2000 = 29.7%
710 18 pixels	1959/2000 = 97.95%	17/2000 = 0.85	24/2000 = 1.2%	136/2000 = 6.8%	29/2000 = 1.45%	1835/2000 = 91.75%
*534 12 pixels	1842/2000 = 92.1%	138/2000 = 6.9%	20/2000 = 1.0%	1351/2000 = 67.55%	91/2000 = 4.55%	558/2000 = 27.9%
1587 11 pixels	597/2000 = 29.85%	25/2000 = 1.25%	1378/2000 = 68.9%	1972/2000 = 98.6%	24/2000 = 1.2%	4/2000 = 0.2%
561 16 pixels	1949/2000 = 97.45%	17/2000 = 0.85%	34/2000 = 1.7%	29/2000 = 1.45%	14/2000 = 0.7%	1957/2000 = 97.85%
435 5 pixels	1813/2000 = 90.65%	16/2000 = 0.8%	171/2000 = 8.55%	117/2000 = 5.85%	15/2000 = 0.75%	1868/2000 = 93.4%
102 7 pixels	1870/2000 = 93.5%	29/2000 = 1.45%	101/2000 = 5.05%	42/2000 = 2.1%	27/2000 = 1.35%	1931/2000 = 96.55%
193 24 pixels	1979/2000 = 98.95%	4/2000 = 0.2%	17/2000 = 0.85%	52/2000 = 2.6%	8/2000 = 0.4%	1940/2000 = 97.0%
*1087 11 pixels	1145/2000 = 57.25%	30/2000 = 1.5%	825/2000 = 41.25%	1914/2000 = 95.7%	45/2000 = 2.25%	41/2000 = 2.05%
**1143 27 pixels	1970/2000 = 98.5%	29/2000 = 1.45%	1/2000 = 0.05%	1958/2000 = 97.9%	42/2000 = 2.1%	0
903 4 pixels	1646/2000 = 82.3%	25/2000 = 1.25%	329/2000 = 16.45%	3/2000 = 0..15%	35/2000 = 1.75%	1962/2000 = 98.1%
*626 6 pixels	1898/2000 = 94.9%	49/2000 = 2.45%	53/2000 = 2.65%	1276/2000 = 63.8%	34/2000 = 1.7%	690/2000 = 34.5%
*1382 21 pixels	1946/2000 = 97.3%	52/2000 = 2.6%	2/2000 = 0.1%	1490/2000 = 74.5%	37/2000 = 1.35%	473/2000 = 23.65%
180 14 pixels	1895/2000 = 94.75%	45/2000 = 2.25%	60/2000 = 3.0%	580/2000 = 29.0%	25/2000 = 1.25%	1395/2000 = 69.75%
*484 7 pixels	1358/2000 = 67.9%	85/2000 = 4.25%	557/2000 = 27.85%	1894/2000 = 94.7%	55/2000 = 2.75%	51/2000 = 2.55%
*135 8 pixels	1557/2000 = 77.85%	102/2000 = 5.1%	341/2000 = 17.05%	1828/2000 = 91.4%	62/2000 = 3.1%	110/2000 = 5.5%
*49 5 pixels	1410/2000 = 70.5%	71/2000 = 3.55%	519/2000 = 25.95%	1884/2000 = 94.2%	55/2000 = 2.75%	61/2000 = 3.05%
*407 11 pixels	1639/2000 = 81.95%	104/2000 = 5.2%	257/2000 = 12.85%	1915/2000 = 95.65%	73/2000 = 3.65%	14/2000 = 0.7%
*523 15 pixels	1960/2000 = 98.0%	30/2000 = 1.5%	10/2000 = 0.5%	1519/2000 = 75.95%	40/2000 = 2.0%	441/2000 = 22.05%
794 4 pixels	1046/2000 = 52.3%	12/2000 = 0.6%	942/2000 = 47.1%	864/2000 = 43.2%	9/2000 = 0.45%	1127/2000 = 56.35%
1210 10 pixels	666/2000 = 33.3%	52/2000 = 2.6%	1282/2000 = 64.1%	1929/2000 = 96.45%	41/2000 = 2.05%	30/2000 = 1.5%
907 10 pixels	1663/2000 = 83.15%	40/2000 = 2.0%	297/2000 = 14.85%	389/2000 = 19.45%	30/2000 = 1.5%	1581/2000 = 79.05%

Table C.1: Results of classifying the second 2000 items of the 0s and 1s training sets

using the first 3000 items of the same sets as the training data. The best performers are marked with a double asterisk and the good 'partial' recognisers are marked with a single asterisk.

Appendix D The full Classification Incidence Matrix

Table D1 is comprised 76 rows x 200 columns spread over ten pages. Each row represents a classification window and each column a test object. The body of the table gives the number of the training image involved in the classification. The topmost column on each page gives the test object numbers, and the classification window numbers are repeated in the leftmost column on each page.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
15	1	8	25	62	82	4	96	82	4	96	3	32	9	32	66	5	12	79	63	19
16	2	26	66	73	2	4	46	96	4	46	4	83	61	9	66	66	37	61	41	18
17	66	9	66	1	26	4	94	96	75	20	61	83	94	31	66	66	75	66	40	18
18	73	7	69	2	7	4	18	75	4	9	9	47	9	14	66	66	75	54	18	18
24	1	75	12	33	3	4	15	3	4	3	15	3	20	66	4	3	12	54	10	19
31	75	38	75	50	17	55	9	94	9	34	5	3	5	83	5	5	27	5	13	70
35	1	22	49	73	41	4	25	3	18	66	18	61	19	41	4	5	19	79	5	44
37	23	22	17	73	94	4	6	19	62	44	75	54	61	40	19	19	62	82	10	75
44	9	9	50	40	55	1	47	3	4	4	1	3	1	29	5	3	80	78	3	44
45	3	9	46	73	15	4	38	3	44	100	61	3	4	3	5	5	5	3	5	44
52	13	71	7	8	7	47	19	94	94	94	61	3	25	9	21	21	90	94	3	3
53	2	1	2	9	3	47	55	3	4	3	27	3	94	24	2	3	80	27	3	5
54	27	9	9	17	3	4	47	3	4	100	27	3	94	27	40	5	80	27	3	74
57	45	8	86	14	62	4	61	82	77	82	75	10	75	1	66	4	62	66	82	62
58	66	6	40	2	13	2	6	3	4	69	22	3	2	3	54	5	76	54	66	81
62	18	79	91	8	21	47	21	21	4	21	61	25	61	5	5	21	12	94	21	94
72	26	35	21	94	15	47	94	94	4	21	97	8	44	27	5	6	80	94	8	5
73	1	96	7	94	3	47	85	29	4	3	4	3	4	3	5	5	80	15	3	5
83	29	1	14	22	7	47	22	100	4	7	22	45	22	45	21	5	80	45	45	5
84	1	1	47	47	10	66	85	22	32	82	26	6	25	45	73	6	80	10	10	5
85	1	1	47	46	84	10	10	84	4	84	5	64	5	78	66	5	5	47	78	61
122	44	1	32	1	26	4	84	10	73	3	44	30	98	3	3	3	15	15	3	5
123	92	1	21	1	22	61	1	15	4	82	26	52	61	3	61	74	82	3	53	21
132	1	1	14	28	25	84	25	11	1	3	81	75	98	3	3	3	45	80	80	6
133	92	1	21	92	89	1	14	85	73	82	73	53	44	3	61	3	27	91	53	6
136	1	92	22	94	22	22	46	82	70	98	77	3	5	3	73	73	77	46	82	100
143	1	1	92	29	77	36	87	82	4	4	1	75	44	3	66	3	27	82	48	47
146	70	70	2	82	85	85	75	22	4	82	26	1	4	1	74	65	15	16	80	61
153	1	29	92	92	70	94	84	15	4	3	73	2	15	3	16	3	29	15	86	82
154	29	1	92	94	61	32	30	82	86	3	86	3	82	11	21	24	92	45	86	15
163	2	27	29	2	80	40	46	15	1	4	4	3	15	3	56	3	29	15	56	6
164	1	1	29	70	27	100	1	15	61	22	4	3	80	3	46	46	23	15	86	80
173	1	15	29	2	7	26	92	80	15	10	47	3	47	3	46	46	92	58	78	58
174	2	2	2	86	46	100	100	49	3	82	4	67	49	3	17	76	92	82	3	46
183	44	27	77	49	52	30	27	52	15	28	4	3	4	3	73	49	92	83	61	46
184	1	1	1	17	7	67	46	76	45	3	82	46	19	3	61	61	50	91	94	76
185	1	1	1	73	17	100	100	16	3	25	59	25	72	6	11	16	50	29	12	18
186	82	80	99	2	6	10	6	29	4	25	88	25	86	3	23	11	90	29	28	6
187	1	2	3	75	76	100	75	50	45	26	45	2	4	3	61	61	82	30	25	4
193	44	22	45	28	11	33	11	80	4	28	80	79	83	3	73	50	92	88	13	8
194	1	1	1	17	29	6	45	6	19	92	53	28	49	3	9	26	92	5	8	78
195	1	37	97	100	70	6	24	6	53	3	53	46	88	3	88	17	17	6	25	6
196	24	37	24	2	97	6	6	6	49	3	49	10	40	3	75	17	88	6	25	6
197	26	47	3	75	41	25	75	25	45	71	45	61	34	26	61	61	25	50	58	19
198	26	1	76	38	30	19	46	26	49	56	45	65	49	11	61	61	29	18	3	19
204	1	43	35	1	47	6	85	6	53	46	76	46	66	41	44	44	1	5	20	6
205	1	77	36	21	77	1	8	6	4	3	19	3	4	3	17	36	78	6	29	18
206	92	84	77	43	77	8	2	6	29	41	29	10	29	3	17	17	83	6	2	6
213	5	85	46	36	35	85	1	86	76	25	15	3	76	3	49	3	50	12	50	40
214	1	1	47	62	91	1	82	42	53	39	44	25	50	3	4	16	28	16	39	16
218	62	1	62	35	46	19	2	6	34	1	16	61	45	55	17	61	13	29	15	16
223	6	2	35	37	77	30	30	6	49	4	57	25	50	12	19	12	26	6	13	19
224	1	2	35	27	27	50	62	76	49	56	49	3	49	3	74	16	28	16	39	16
225	1	2	36	27	36	50	50	50	19	13	19	3	49	3	5	6	16	6	13	5
228	62	2	62	2	46	6	66	61	61	3	20	1	45	55	17	5	15	16	15	19
232	76	1	36	36	16	15	3	54	49	4	57	25	49	79	19	16	5	41	3	41
234	69	35	27	36	35	16	16	69	87	94	49	3	49	3	28	12	5	12	5	42
236	66	1	36	2	35	1	1	20	16	50	15	3	49	25	5	5	40	24	13	5
237	36	2	96	36	60	16	1	5	46	50	45	6	45	3	17	17	16	40	13	16
241	26	1	1	86	26	35	14	26	4	50	26	50	50	46	5	5	16	16	79	10
242	50	1	62	2	62	35	20	26	5	100	16	79	50	76	5	79	16	76	12	12
243	35	43	62	62	62	66	35	6	5	33	42	10	49	10	16	5	40	41	20	92
245	1	1	35	2	43	2	87	62	15	82	5	3	49	3	16	16	16	35	19	16
248	1	2	36	2	9	40	2	20	49	40	49	3	49	3	6	6	5	13	6	16
247	36	36	46	36	46	13	2	3	34	3	40	3	49	39	6	6	16	16	6	26
248	36	2	46	14	46	39	26	20	57	1	50	26	45	41	6	6	1	16	6	45
251	76	1	26	2	26	19	19	26	57	17	69	20	19	72	44	44	20	20	46	16
252	50	1	86	2	26	66	41	26	6	36	16	3	50	46	11	79	16	20	79	3
253	50	1	96	66	26	46	50	36	60	4	15	10	20	10	60	60	16	12	10	16
255	1	1	67	67	67	67	67	67	29	50	49	3	4	6	49	40	16	13	39	21
261	36	1	16	16	29	5	50	35	46	33	15	20	46	46	44	44	16	20	44	20
262	41	2	96	96	16	45	12	41	50	4	4	3	4	50	36	50	19	26	10	10
264	41	1	16	16	26	26	16	12	36	37	77	3	17	3	66	24	12	26	12	6
265	26	1	69	1	26	1	26	26	55	57	16	3	100	3	66	42	16	69	16	22
266	12	1	6	30	62	1	1	13	40	40	3	3	16	16	16	3	16	5	16	24
268	26	24	62	66	56	12	46	15	4	3	15	22	31	12	36	4	16	24	16	1

	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
15	60	12	12	48	48	56	66	5	34	6	29	19	56	79	48	95	96	75	62	79
16	4	56	56	34	66	10	56	63	55	6	74	28	48	48	48	66	96	19	66	48
17	9	56	7	43	54	10	54	5	49	6	26	7	56	54	16	29	66	43	95	64
18	75	9	7	7	76	54	54	46	25	6	29	2	78	56	54	45	56	7	1	76
24	6	12	12	81	56	55	24	10	72	6	29	12	72	56	24	3	48	1	3	5
31	20	27	3	73	5	73	18	5	55	6	18	6	20	18	10	5	73	75	7	5
35	9	54	54	80	80	20	37	5	16	6	6	61	16	98	19	70	98	7	41	4
37	82	19	62	43	82	20	10	74	10	26	26	82	10	10	16	5	5	3	26	54
44	6	12	12	55	4	1	10	5	20	6	97	4	10	10	44	100	100	3	3	5
46	6	54	54	80	44	59	10	5	16	6	4	61	10	5	44	66	28	4	41	44
52	20	13	61	16	21	3	25	21	20	6	29	13	10	18	10	21	21	8	7	21
53	6	62	62	27	21	3	10	5	34	13	8	12	15	6	24	65	65	3	3	5
54	6	12	12	29	29	6	10	10	10	13	16	1	10	10	44	28	65	75	3	29
57	7	62	75	66	66	62	10	5	15	6	2	26	43	10	2	66	5	76	77	94
58	9	3	9	9	3	54	10	5	15	16	7	77	15	10	7	5	66	26	77	65
62	20	12	3	78	21	9	78	21	10	75	5	12	10	16	10	21	21	6	30	21
72	30	12	62	94	94	6	6	5	10	75	5	30	6	6	6	22	100	6	30	5
73	6	12	12	94	15	66	10	5	16	53	5	60	9	6	24	14	51	6	6	5
83	6	75	29	66	45	45	45	5	10	75	5	21	45	24	24	7	16	6	60	5
84	3	3	3	25	65	3	10	4	3	9	33	4	10	16	16	5	5	75	24	5
86	82	65	82	94	77	6	54	5	54	9	23	5	82	81	82	5	5	44	9	4
122	6	75	15	10	5	75	16	47	3	6	5	15	16	10	3	10	55	47	75	66
123	6	53	76	76	76	60	3	58	3	6	51	47	3	5	3	51	51	6	6	62
132	24	15	15	5	5	15	24	47	16	6	51	15	16	51	3	10	64	6	15	66
133	10	53	47	76	76	76	3	5	60	6	51	47	3	51	3	51	51	66	60	4
136	66	27	15	5	51	62	6	73	99	62	6	15	16	28	16	5	5	28	90	25
143	10	10	10	62	44	56	3	5	6	10	51	76	3	51	3	74	74	62	62	66
146	10	46	46	9	25	25	10	7	6	9	6	7	11	9	6	6	6	25	10	26
152	7	76	45	15	46	96	3	5	55	3	22	17	24	62	24	74	74	63	51	46
154	11	10	15	73	71	12	3	5	6	24	33	26	24	5	24	51	71	5	45	77
163	29	76	46	56	62	84	3	45	6	3	44	5	3	39	3	74	74	46	30	12
164	77	60	62	46	60	13	16	5	6	3	100	51	24	74	6	51	71	56	45	60
173	29	47	45	56	29	84	73	6	6	5	75	15	3	29	3	63	74	47	30	64
174	79	29	29	71	29	19	13	74	76	3	33	15	26	21	76	50	71	66	22	21
183	10	30	55	56	29	76	5	44	6	3	22	15	3	29	3	30	73	62	30	5
184	12	62	29	22	62	13	13	5	6	3	45	45	26	77	6	5	51	56	48	61
186	25	77	47	8	62	40	5	61	76	6	8	49	30	49	6	5	44	60	8	16
196	48	60	46	96	60	64	66	5	46	76	8	30	6	49	6	5	8	100	60	4
197	60	1	1	15	61	45	66	67	77	60	5	66	25	5	10	5	61	57	13	45
198	10	11	1	46	67	67	3	48	3	6	8	15	3	39	3	6	10	28	78	5
194	12	48	46	28	23	13	12	5	6	5	44	15	26	10	6	5	67	8	13	5
195	24	2	2	61	11	12	12	5	5	5	8	45	6	8	6	56	5	18	1	76
196	24	1	37	61	60	30	51	5	1	37	8	49	6	45	6	45	61	18	1	66
197	25	25	56	45	17	49	30	64	47	6	5	5	6	16	39	5	36	61	25	66
198	29	26	3	45	44	99	13	16	16	29	5	5	6	45	57	5	66	99	3	57
204	48	62	45	17	37	26	48	45	46	5	45	45	48	10	6	10	51	8	28	13
205	24	46	1	36	15	5	13	45	6	66	6	45	6	5	39	55	66	16	13	49
206	24	37	24	66	63	5	51	5	6	39	5	49	6	45	6	45	61	40	61	4
213	10	59	55	29	60	25	10	76	6	3	16	15	50	10	50	10	10	40	48	49
214	39	46	46	2	5	5	39	49	76	13	45	44	76	10	16	10	33	50	13	50
218	19	41	91	24	29	24	99	29	6	39	12	29	6	16	47	57	54	40	3	29
223	13	28	70	2	5	56	48	50	44	25	60	48	25	8	25	49	29	19	12	5
224	20	79	46	2	5	25	24	15	50	3	45	45	20	10	20	5	10	6	13	5
225	20	50	77	55	40	6	46	5	24	13	5	45	46	45	45	16	55	5	25	5
226	19	6	39	45	49	16	67	54	29	20	50	39	16	20	51	15	29	16	1	61
232	13	28	27	11	42	25	24	15	44	12	63	8	13	8	13	5	18	16	25	16
234	26	13	27	77	5	25	14	67	44	13	44	63	3	8	10	49	19	19	13	12
236	13	39	30	6	5	13	61	15	1	13	5	15	13	15	10	16	24	5	50	5
237	20	6	20	57	5	51	51	20	67	16	67	18	20	24	51	15	5	20	6	16
241	16	16	8	15	16	12	74	44	13	16	15	16	26	44	16	16	8	16	74	5
242	20	16	12	35	16	16	16	44	13	39	15	8	39	44	12	16	6	40	26	5
243	13	20	26	42	40	26	67	44	26	13	67	8	13	44	10	40	6	40	12	5
245	13	12	67	8	16	25	10	15	10	3	15	15	10	15	10	16	16	16	6	5
246	13	13	67	30	39	51	51	16	25	13	6	15	13	49	10	16	15	16	6	5
247	6	13	16	57	20	51	51	5	30	15	8	24	16	24	51	49	24	16	6	13
248	6	13	13	45	24	24	47	54	29	13	6	29	13	24	51	6	6	52	6	16
251	16	16	16	10	19	10	60	44	10	16	56	16	6	44	20	16	6	19	60	5
252	10	16	16	16	19	10	26	44	26	6	16	6	6	44	6	16	6	15	26	5
253	46	16	12	36	42	10	26	59	10	10	66	42	10	44	10	16	6	16	10	26
255	13	12	16	40	77	43	25	6	12	14	6	6	25	49	10	15	6	16	36	54
261	49	50	16	26	21	49	74	35	10	6	29	36	20	44	20	6	6	29	45	97
263	10	6	16	6	66	14	21	15	44	10	15	15	10	60	50	5	6	5	12	49
264	6	26	26	16	66	14	14	26	27	60	29	29	49	49	44	15	6	77	10	66
265	6	12	12	60	66	76	76	26	27	41	15	15	41	45	9	6	15	5	6	24
266	6	12	27	15	5	76	14	26	76	41	29	6	73	15	76	6	6	44	14	19
268	6	2	2	27	1	6	6	23	6	6	29	26	10	27	6	27	46	5	6	66

	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
15	79	72	13	77	34	1	7	75	75	75	8	48	77	48	78	32	50	56	56	77
16	96	48	66	33	75	2	8	8	77	8	7	48	4	9	78	100	9	79	56	77
17	10	54	26	75	2	73	8	80	61	43	8	48	75	48	10	100	20	66	20	77
18	54	78	19	62	25	77	12	97	75	43	42	8	9	9	10	56	9	56	25	75
24	83	72	20	34	72	17	8	8	9	34	8	72	9	9	10	55	3	56	63	6
31	73	10	8	50	10	8	8	8	8	2	8	10	7	9	10	7	7	5	5	6
35	79	10	20	77	80	76	7	7	7	7	7	80	9	9	18	25	3	24	5	6
37	10	10	75	17	15	73	14	66	7	8	7	10	2	9	43	21	9	13	13	82
44	10	96	50	75	37	75	3	75	75	12	2	10	9	9	86	10	13	86	86	6
45	10	81	9	21	80	75	7	7	7	7	2	10	9	9	18	10	3	57	57	6
52	10	10	9	13	74	13	8	8	13	34	8	10	14	9	10	9	7	21	5	20
53	10	10	14	9	8	53	8	9	9	96	8	10	7	9	10	9	9	5	86	6
54	10	10	9	75	10	75	14	7	75	12	8	10	9	9	10	55	6	10	10	6
57	10	10	44	17	66	26	76	2	2	54	77	10	19	7	43	83	7	86	19	2
58	54	54	82	80	54	75	26	26	75	78	77	10	9	9	43	58	25	5	5	9
62	10	10	5	3	74	7	8	8	75	7	8	10	5	8	10	16	10	21	81	8
72	10	10	8	3	10	7	8	8	68	7	5	10	9	9	10	9	9	5	81	60
73	10	24	14	3	10	7	32	7	7	78	8	10	3	9	10	9	6	5	40	9
83	45	45	14	7	10	9	44	45	7	82	8	10	7	9	10	24	8	40	81	8
84	45	10	82	6	10	7	21	23	7	53	4	10	7	9	10	45	3	4	23	82
85	10	54	5	44	10	7	44	44	7	9	8	10	9	9	82	8	30	5	44	9
122	55	18	5	15	55	9	5	5	30	30	100	8	55	9	55	5	30	51	51	7
123	15	8	5	45	53	7	51	5	3	9	83	10	55	9	15	78	24	47	58	60
132	55	18	5	5	6	93	28	61	18	38	83	8	5	7	10	5	9	47	47	23
133	45	8	15	15	53	21	51	73	15	30	83	8	15	7	76	10	24	5	78	75
138	2	68	100	100	8	9	5	75	8	46	5	82	4	9	2	8	7	4	4	75
142	3	3	76	91	53	80	51	73	80	30	28	8	10	21	10	6	21	76	98	80
148	21	10	55	9	81	9	8	8	55	91	8	82	95	9	10	10	91	26	35	87
153	90	90	11	91	53	75	14	69	11	31	28	53	76	9	10	10	7	76	84	91
154	3	24	8	8	53	7	73	30	53	31	33	53	6	30	10	6	7	76	83	75
163	6	90	76	45	8	74	83	3	80	31	28	10	78	9	76	91	95	53	88	80
164	3	6	17	17	53	14	48	48	31	30	8	86	4	9	48	80	30	91	77	91
173	6	6	52	45	53	9	8	30	6	9	8	10	91	30	91	9	9	10	80	91
174	3	8	52	47	8	7	28	8	84	71	28	10	59	7	9	45	9	1	15	14
183	83	81	80	52	6	9	18	30	99	9	83	8	9	9	10	39	9	48	15	21
184	3	6	7	9	4	75	20	25	19	7	68	6	59	9	48	6	21	47	49	100
185	3	16	23	23	16	9	8	8	72	9	6	6	23	7	10	6	9	40	53	14
186	88	26	23	23	99	7	8	6	26	73	5	50	52	9	46	50	9	26	42	64
187	1	81	7	59	47	9	74	5	52	52	8	82	59	7	2	10	7	79	39	9
193	98	6	52	91	6	7	18	82	47	7	83	6	7	9	39	10	7	6	89	24
194	3	8	88	59	6	9	8	25	47	7	8	8	83	9	10	6	9	78	80	9
195	5	78	52	47	8	74	25	28	83	17	8	6	33	7	24	8	44	83	47	14
196	13	63	75	23	1	84	5	46	26	17	8	82	17	7	25	6	7	40	50	14
197	99	98	14	52	47	7	8	5	65	46	5	51	52	7	75	24	7	16	50	7
198	3	5	7	9	5	9	51	5	91	59	66	6	9	9	78	28	75	51	6	23
204	5	5	52	25	6	9	8	8	78	35	8	8	78	9	10	6	9	44	15	25
205	3	5	52	23	5	48	3	3	80	23	8	6	80	9	10	6	23	17	78	14
206	41	41	9	75	47	56	5	2	83	46	5	10	97	9	30	45	9	49	60	14
213	10	6	52	25	6	9	18	18	99	9	8	8	29	7	10	39	9	11	44	37
214	79	78	87	82	45	9	8	13	78	25	8	76	78	7	10	8	83	17	88	81
216	27	47	53	2	29	9	5	15	78	9	54	8	44	7	78	29	8	51	3	81
223	10	6	53	31	46	9	12	20	86	7	8	6	83	9	10	39	47	11	99	9
224	92	50	53	82	45	35	8	13	31	42	45	26	31	7	10	8	7	2	15	7
225	88	45	8	2	87	7	8	8	83	58	45	28	9	7	79	45	43	29	86	9
228	87	87	35	35	87	9	15	8	92	8	5	12	9	7	10	29	8	51	51	83
233	74	25	53	99	46	43	5	16	47	47	19	25	47	7	48	8	84	81	1	9
234	88	8	53	47	47	9	12	8	23	43	28	28	31	7	14	44	9	82	88	9
238	25	1	2	2	47	92	8	8	9	9	5	25	53	7	10	45	47	88	11	9
237	47	87	2	88	87	9	87	87	2	7	13	20	77	7	10	29	14	40	47	38
241	74	16	9	24	26	35	15	49	27	68	15	10	9	21	10	58	88	35	35	24
242	42	12	99	99	28	43	15	89	83	9	19	6	9	33	39	45	90	4	25	27
243	18	12	36	9	25	78	18	18	9	7	18	13	31	38	44	44	90	89	46	7
245	3	87	43	44	87	58	18	18	80	78	8	6	9	7	10	45	33	20	20	7
246	73	13	2	47	87	7	8	18	9	9	8	13	2	7	10	10	47	77	31	47
247	25	30	14	2	30	35	8	8	14	9	8	39	46	7	51	31	10	51	31	92
248	47	24	14	14	2	9	8	8	14	9	8	39	9	9	51	24	10	51	15	4
251	80	12	33	33	28	32	15	12	45	1	8	10	9	9	8	48	77	42	17	14
252	80	12	7	58	26	96	15	15	45	9	8	10	7	7	12	24	33	15	89	72
253	12	10	36	58	12	9	19	18	7	7	8	10	7	7	10	44	43	89	89	27
255	14	26	36	27	67	90	8	18	88	27	8	13	7	9	32	88	32	39	57	88
261	45	12	38	7	28	31	39	58	41	21	29	10	7	9	10	45	31	15	5	12
263	93	12	7	7	13	9	15	8	45	7	29	41	7	9	50	58	81	13	5	50
264	28	26	38	33	26	9	40	19	58	58	15	25	7	95	49	10	79	16	16	35
265	14	41	80	58	12	9	86	1	24	58	15	41	9	9	88	10	35	24	6	86
266	14	3	33	49	26	9	11	11	86	7	5	12	70	7	79	49	8	49	10	9
268	10	10	25	58	6	7	8	29	14	62	55	2	58	9	38	27	46	10	10	49

	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
15	62	7	9	8	7	83	79	1	48	36	96	48	9	9	82	12	62	93	95	82
16	62	1	6	8	8	48	48	26	48	14	83	75	76	4	66	12	66	13	13	11
17	77	7	2	8	8	43	54	26	54	94	94	16	16	2	2	12	16	20	99	11
18	9	75	76	8	8	78	78	4	78	78	78	78	9	75	2	12	16	26	33	11
24	65	8	34	34	34	56	56	13	72	79	32	24	3	12	3	12	14	20	13	82
31	22	8	50	8	8	10	5	17	78	73	73	10	6	8	83	21	17	13	13	78
35	26	22	3	8	8	80	80	26	16	79	79	80	3	75	3	12	49	49	49	41
37	14	82	62	8	8	10	10	26	10	78	16	82	54	62	73	82	26	19	68	53
44	97	8	3	9	8	10	5	21	10	79	55	3	41	3	3	12	13	74	49	20
45	26	4	3	8	8	16	5	8	10	79	79	80	3	81	3	12	3	49	49	24
52	94	29	13	5	8	8	21	81	10	21	73	10	20	12	8	12	60	60	7	68
53	81	8	9	3	8	18	3	61	10	55	21	20	6	53	6	12	13	13	13	53
54	1	7	3	3	6	6	24	44	10	79	15	53	6	12	6	12	3	14	21	53
57	26	82	1	8	7	82	10	13	68	78	35	94	81	82	1	82	26	26	44	10
58	14	5	1	7	8	10	10	26	54	78	10	54	78	75	1	12	49	26	19	54
62	29	5	12	21	21	10	21	61	10	21	21	10	20	12	75	12	17	17	16	68
72	33	8	30	5	8	8	8	91	10	8	8	8	6	30	75	30	80	18	16	51
73	56	14	6	3	3	3	5	17	10	25	21	24	8	7	53	60	60	16	19	68
83	33	45	9	45	7	45	5	61	10	27	45	45	6	9	6	60	60	12	12	84
84	33	44	75	100	6	6	28	8	10	10	45	6	24	82	6	52	12	20	19	84
85	66	66	30	77	8	54	77	44	82	10	47	3	85	5	6	12	12	19	4	1
122	5	51	9	75	55	75	6	11	5	5	5	6	6	5	6	13	81	16	94	45
123	58	51	60	6	15	3	45	76	10	11	32	80	6	76	6	12	98	98	13	6
132	5	51	18	15	5	75	10	10	18	11	8	80	6	5	6	19	19	13	81	45
133	11	51	60	6	60	3	10	10	76	11	8	80	6	76	6	94	94	12	12	60
136	22	28	100	15	16	62	100	10	16	11	22	16	53	77	10	16	98	98	98	13
143	17	51	60	62	66	3	51	62	10	52	75	80	6	17	6	70	94	27	27	6
148	22	9	87	75	75	10	25	25	25	25	54	75	10	75	91	98	13	38	61	10
153	17	5	76	94	15	3	63	46	76	52	47	80	82	15	53	69	12	13	13	11
154	21	5	6	45	3	24	74	2	6	2	87	48	78	45	66	69	69	92	92	11
163	21	76	91	69	11	95	39	47	76	67	77	6	11	15	53	69	63	73	73	60
164	21	14	17	69	3	74	74	4	55	23	48	6	4	23	82	46	69	45	47	11
173	46	74	7	83	5	3	39	48	7	87	84	6	11	80	80	63	63	73	73	7
174	17	73	9	5	5	78	77	48	17	23	69	76	4	82	60	58	6	88	28	29
183	91	74	21	78	78	81	39	5	21	58	54	6	92	45	6	30	18	19	16	45
184	46	77	7	61	5	78	77	76	62	97	5	5	11	44	76	44	16	40	42	93
185	23	74	12	5	5	5	12	16	68	92	5	5	76	47	47	12	76	51	40	91
186	23	49	9	15	58	63	13	17	44	59	59	77	77	46	82	12	50	45	40	91
187	44	79	71	99	16	45	67	1	2	1	59	1	1	66	66	45	50	49	45	92
193	97	67	7	27	65	6	39	31	3	78	66	5	6	45	5	16	41	6	50	91
194	29	71	9	10	5	49	10	76	61	84	5	5	1	46	6	13	63	39	5	63
195	47	12	12	78	58	16	13	9	55	64	84	5	76	6	47	12	80	5	5	91
196	46	73	9	99	30	63	45	76	55	67	84	13	37	44	44	73	63	98	40	65
197	46	79	84	13	15	5	25	59	23	85	21	47	6	47	82	12	29	16	57	70
198	46	16	55	100	78	5	80	55	77	76	85	5	3	13	91	12	16	13	16	91
204	35	71	27	39	5	77	10	62	99	90	90	5	6	77	12	12	44	16	16	35
205	91	9	77	67	5	77	13	9	95	38	90	5	58	6	29	12	62	94	5	23
206	46	73	36	58	99	63	67	9	51	77	90	41	76	2	80	12	63	49	18	35
213	47	24	38	6	16	55	6	17	44	66	36	5	6	26	6	16	16	73	16	9
214	35	62	60	45	45	55	12	11	45	90	84	50	6	92	45	12	63	12	16	35
218	73	29	44	24	6	47	27	90	26	90	60	1	38	16	93	16	13	16	16	14
223	43	10	2	48	10	55	48	32	8	76	2	44	48	92	63	12	41	41	41	7
224	37	14	90	45	45	55	39	77	45	97	2	50	92	26	67	10	42	16	12	27
225	37	1	84	93	93	45	24	55	8	76	23	50	66	68	45	20	78	40	40	36
228	90	29	37	47	8	45	81	76	47	76	23	22	98	13	19	13	13	16	20	46
233	47	8	27	48	48	11	48	32	83	86	84	25	48	79	79	41	12	26	16	37
234	36	44	47	45	45	77	10	76	83	70	60	18	6	46	67	12	12	26	26	35
236	80	8	43	48	47	45	48	100	8	93	27	50	51	79	27	1	12	16	19	43
237	53	29	36	47	47	29	82	48	100	48	37	50	47	51	24	13	13	20	18	35
241	9	44	14	48	58	58	46	31	56	24	2	12	41	93	8	16	16	16	16	75
242	52	44	14	26	44	86	79	99	66	86	50	44	44	67	79	12	41	16	12	43
243	9	45	14	44	44	11	44	7	66	66	83	12	44	79	77	16	19	18	16	36
245	29	45	86	48	79	24	14	60	6	14	80	3	45	10	86	12	12	40	40	43
246	23	29	44	87	47	45	67	44	87	93	27	6	51	79	27	62	16	5	16	35
247	27	8	55	87	82	45	32	46	17	43	10	6	81	51	57	13	13	18	18	35
248	21	29	55	47	99	45	36	70	61	22	10	17	67	47	57	12	12	13	29	34
251	9	44	14	79	58	79	46	60	77	35	28	16	89	93	72	16	16	18	16	36
252	52	79	14	46	46	1	46	45	86	66	50	44	44	13	58	6	6	12	19	22
253	65	45	27	73	44	86	73	7	29	66	50	10	44	79	79	12	16	16	19	43
255	93	24	66	46	10	49	3	27	6	10	66	50	44	10	49	12	28	16	16	97
261	30	44	14	48	46	4	46	30	1	43	9	10	46	73	48	16	70	13	13	36
263	7	15	44	50	79	24	50	7	60	66	41	50	50	79	79	16	16	16	16	52
264	58	58	95	79	44	66	50	33	29	58	58	3	49	44	49	26	26	26	26	94
265	58	24	49	48	10	59	3	33	29	58	49	3	6	3	49	12	12	12	12	37
266	58	15	55	3	15	45	3	22	66	10	7	16	79	3	51	12	12	8	8	97
268	12	2	82	63	49	57	16	22	93	22	46	16	47	48	79	12	2	15	98	36

	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
15	27	83	27	47	65	84	23	87	95	30	30	24	30	25	25	74	18	93	36	36
16	14	20	14	83	9	20	55	25	95	90	53	31	31	13	25	74	34	93	35	36
17	31	32	3	55	83	55	22	53	95	53	53	53	30	26	25	31	34	92	38	36
18	24	54	20	25	25	84	100	52	95	28	95	20	24	25	25	29	8	26	37	36
24	38	69	46	20	20	55	69	49	32	87	100	55	26	61	25	94	24	25	35	36
31	11	20	83	55	55	5	49	90	12	26	12	26	15	25	25	49	98	34	38	36
35	34	25	20	20	3	20	27	80	95	100	90	25	11	75	25	45	34	48	38	36
37	47	20	7	20	7	24	23	89	28	25	28	53	23	29	25	45	34	17	37	57
44	66	32	20	3	68	16	29	25	90	28	62	55	55	3	25	27	34	33	43	36
46	54	27	20	3	3	59	29	25	95	55	70	55	42	3	25	44	89	25	38	63
52	79	20	6	91	68	7	21	26	84	12	62	9	18	7	25	65	34	96	38	36
53	86	20	20	20	68	21	15	86	70	12	41	9	55	9	25	65	54	78	35	56
54	24	20	13	13	68	20	24	28	41	30	90	55	36	82	25	65	54	96	35	36
57	29	10	9	19	20	5	21	25	26	24	26	25	19	49	25	52	34	46	38	36
58	54	54	8	20	20	13	14	28	18	29	16	26	20	25	25	82	34	29	42	35
62	22	20	6	55	62	10	10	84	22	30	22	21	35	9	25	1	36	48	37	58
72	22	24	3	20	16	8	21	26	26	30	26	8	98	9	25	47	37	48	36	36
73	84	20	75	20	13	18	91	26	35	12	28	22	44	7	25	98	89	89	72	37
83	100	12	75	20	12	24	5	88	26	30	22	35	1	7	25	47	46	78	72	37
84	1	13	3	13	13	24	18	26	88	84	100	28	73	82	25	25	78	58	37	36
85	1	12	52	20	20	82	97	28	70	84	77	25	66	30	25	25	78	78	37	56
122	24	54	10	19	16	3	68	91	26	9	78	44	56	24	25	92	49	48	43	37
123	24	65	66	98	4	52	76	83	33	24	69	44	56	3	25	92	49	34	37	20
132	7	54	68	19	12	3	68	8	63	46	73	21	56	21	25	92	50	44	37	37
133	3	87	6	13	13	3	11	28	33	77	77	21	56	3	25	94	50	59	37	35
136	87	98	45	3	98	51	73	28	28	3	4	51	28	3	25	28	74	85	37	36
143	3	92	68	92	73	3	55	28	33	62	44	74	74	99	25	92	49	66	37	37
148	11	98	70	45	70	55	97	73	33	45	25	9	26	24	25	27	30	49	37	37
153	24	65	15	73	12	6	21	75	100	2	28	13	74	4	25	29	28	39	37	37
154	45	92	45	3	67	55	59	33	33	2	22	28	78	83	25	27	34	34	37	37
163	29	73	6	83	83	6	21	28	100	2	28	100	26	48	25	29	34	39	37	37
164	29	45	83	65	79	71	55	22	28	2	30	28	83	49	25	30	34	34	38	35
173	29	40	42	79	83	9	7	100	33	6	33	33	26	49	25	44	34	39	43	36
174	2	67	16	13	79	21	17	33	28	2	100	28	78	49	28	48	34	34	38	43
183	93	16	42	83	20	9	17	100	100	92	92	100	50	45	25	92	73	73	43	36
184	1	12	51	67	79	21	17	25	25	50	48	28	49	50	25	48	34	34	43	36
185	91	10	40	98	8	4	17	25	25	50	48	24	40	25	25	48	34	34	36	35
186	91	73	50	16	73	17	100	25	25	29	65	20	49	29	10	48	61	61	38	35
187	91	24	50	66	79	81	80	51	50	29	65	99	83	49	25	50	52	83	37	37
193	29	16	8	79	20	83	61	100	100	92	92	100	92	11	25	48	34	34	37	37
194	97	12	51	67	12	88	17	25	25	92	48	28	49	47	25	48	34	34	43	43
195	91	74	51	13	3	16	2	48	24	65	25	24	39	25	25	48	34	34	43	35
196	91	40	39	98	96	52	59	24	28	50	28	24	49	29	25	47	33	75	43	35
197	91	79	50	79	51	17	21	79	24	29	24	39	49	49	25	92	52	81	35	65
198	91	20	6	20	16	44	100	6	6	6	6	19	57	45	26	6	44	81	36	35
204	35	79	41	67	48	83	21	100	100	47	30	100	53	49	25	30	34	34	38	91
205	23	73	39	13	58	78	55	28	3	15	91	26	15	4	25	48	14	34	77	44
206	91	16	6	13	51	52	59	20	24	50	50	20	4	49	25	35	14	34	77	35
213	9	73	73	41	41	83	17	28	20	50	72	26	26	82	25	20	71	61	37	91
214	35	79	41	16	13	44	21	13	13	5	48	100	34	53	25	48	76	34	43	43
218	91	79	6	18	13	55	17	6	20	20	19	49	25	20	25	24	34	34	38	76
223	7	73	20	41	40	83	74	25	25	50	12	13	18	17	25	28	88	34	38	29
224	70	50	13	41	16	83	21	13	13	50	28	48	16	4	25	13	24	16	43	43
225	35	20	39	41	13	80	21	25	28	5	28	6	19	19	25	50	73	34	38	27
228	76	79	20	13	12	55	61	10	19	20	19	19	63	20	25	24	34	94	53	80
233	43	73	13	40	16	83	14	13	13	16	13	13	13	29	25	26	61	87	37	29
234	35	12	16	40	41	44	21	25	25	40	6	67	8	4	25	12	88	97	37	38
236	35	16	13	16	13	11	33	13	3	40	30	79	86	18	25	12	93	52	38	35
237	96	13	16	13	13	21	22	51	41	18	67	49	34	16	66	13	52	78	14	77
241	43	26	28	6	6	38	87	10	17	18	26	16	41	40	25	16	93	80	37	75
242	36	12	26	40	35	68	24	10	6	16	12	12	12	58	25	12	61	37	38	93
243	43	16	13	28	16	53	29	28	6	18	26	10	8	66	25	26	61	37	7	34
245	35	26	12	12	12	93	61	25	6	28	89	3	49	18	25	12	14	66	70	37
246	7	16	30	16	13	83	17	13	6	16	30	3	19	18	25	12	93	52	38	53
247	34	13	18	13	39	58	61	51	6	18	16	16	19	13	25	28	99	66	36	60
248	34	13	24	13	39	55	61	41	41	29	6	58	77	15	25	39	34	87	37	53
251	37	8	28	98	28	90	36	19	78	16	26	39	57	66	25	16	93	43	36	60
252	36	26	28	2	35	94	75	19	28	16	26	16	44	66	25	16	82	53	36	92
253	37	12	28	30	50	60	74	6	19	28	12	10	8	60	25	12	78	36	64	24
255	84	12	12	12	1	76	34	67	13	18	41	13	93	15	25	26	88	38	38	43
261	61	8	29	35	19	28	11	15	15	70	15	39	39	40	25	39	93	36	38	45
263	84	16	16	1	41	15	55	6	26	16	13	16	39	66	25	16	95	36	94	61
264	37	26	26	16	50	66	52	19	26	8	12	26	39	17	25	26	49	59	36	99
265	94	12	12	18	26	60	74	82	82	15	26	78	3	54	25	12	66	55	37	94
266	94	12	12	18	26	55	5	12	6	5	13	66	3	19	26	12	78	92	36	76
268	36	1	24	2	21	93	4	6	24	15	1	39	44	19	74	2	71	87	36	71

	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
15	83	75	70	56	62	17	54	76	73	58	19	30	35	77	39	46	73	90	56	35
16	75	56	55	83	93	17	66	75	17	79	2	84	34	73	23	73	17	84	29	72
17	10	46	25	78	94	17	66	94	83	56	99	100	58	62	23	76	17	90	74	72
18	33	78	28	10	8	44	83	7	52	56	93	100	18	77	38	76	52	53	65	78
24	89	8	84	65	39	73	58	80	57	79	73	100	4	95	39	33	57	70	66	56
31	84	15	28	77	88	73	81	1	73	73	73	56	15	62	95	33	79	17	83	89
35	15	75	86	83	42	73	78	76	73	79	62	30	58	88	39	73	17	70	48	54
37	15	94	89	78	4	98	83	94	17	72	25	90	78	78	23	73	17	33	100	72
44	55	3	27	40	42	73	39	77	73	72	39	82	40	95	51	73	73	59	99	89
45	80	75	29	93	90	73	58	76	73	72	41	30	35	88	53	73	73	54	36	37
52	99	15	51	48	8	73	20	25	73	59	8	1	29	42	62	14	36	17	66	56
53	77	78	1	40	42	21	13	27	58	86	45	98	7	42	42	33	72	60	99	40
54	55	39	86	29	42	73	20	76	73	36	58	22	56	42	90	73	73	53	99	35
57	6	81	11	10	33	26	58	94	88	72	80	71	71	95	91	77	88	85	100	72
58	77	78	11	10	23	83	35	7	32	72	6	45	96	80	84	80	52	43	24	78
62	98	56	28	40	8	93	3	59	73	4	55	99	5	42	95	22	36	17	89	56
72	54	78	100	40	8	93	53	27	93	65	37	77	46	57	90	44	36	60	54	56
73	54	56	51	25	95	21	60	31	58	57	59	51	97	95	95	58	72	60	69	97
83	63	67	13	25	23	55	80	17	58	57	37	77	66	34	95	22	72	70	54	40
84	67	56	11	65	34	73	3	5	73	57	38	26	35	20	99	33	33	70	54	99
85	80	80	65	10	33	66	78	62	66	57	80	45	78	90	95	73	4	70	80	36
122	50	67	15	73	73	55	53	46	55	16	72	29	55	10	22	6	86	95	48	73
123	50	44	80	4	56	51	60	82	33	4	41	24	66	55	37	6	86	95	41	63
132	50	14	15	57	56	51	15	60	51	19	40	3	42	10	70	6	26	95	16	63
133	50	21	60	81	66	51	60	48	25	98	40	45	56	51	43	6	41	95	82	63
136	99	51	23	48	73	5	85	55	5	72	85	27	88	78	97	73	5	31	54	57
143	48	23	60	40	34	51	60	48	33	98	72	98	56	51	97	51	26	95	99	72
148	9	73	84	1	58	20	40	71	59	81	24	27	24	42	42	99	99	53	58	25
153	26	8	60	13	34	5	60	99	25	52	79	15	58	76	95	15	12	95	96	72
154	55	74	60	19	20	33	53	83	33	84	33	92	7	6	95	45	33	31	52	72
163	24	45	60	18	26	44	53	99	79	72	79	82	7	32	97	51	13	38	94	72
164	71	46	60	34	34	33	53	99	33	84	73	92	7	73	95	8	100	95	94	72
173	34	4	60	34	34	74	53	6	33	72	79	62	55	74	38	8	13	38	84	72
174	71	71	53	34	34	33	17	17	33	72	33	82	14	59	43	6	33	38	6	72
183	48	68	60	3	34	74	53	19	79	72	79	90	55	74	11	51	16	43	97	72
184	71	4	53	87	40	33	77	16	33	72	33	82	17	73	84	67	33	36	4	72
185	71	71	53	3	68	14	74	37	14	41	3	82	72	71	11	28	14	32	4	20
186	61	3	44	68	66	75	20	52	75	72	33	82	57	8	17	3	75	32	17	72
187	61	74	23	41	63	74	72	63	7	72	33	85	20	13	90	74	9	2	81	72
193	7	68	60	3	63	74	10	7	79	72	79	38	48	74	77	82	72	54	78	72
194	7	68	53	31	20	14	71	68	67	72	67	38	14	73	90	67	25	36	4	72
195	16	68	53	76	68	14	71	68	14	41	3	90	68	3	56	28	14	67	59	34
196	61	3	81	86	68	74	20	52	74	72	46	90	57	10	59	46	14	97	59	72
197	61	22	53	72	63	73	72	63	75	72	14	90	86	9	17	15	14	53	6	72
198	61	22	23	79	63	7	72	78	2	63	51	70	68	75	58	74	75	53	39	72
204	84	68	53	78	34	100	10	59	100	72	67	38	81	68	90	67	100	54	59	72
205	38	68	4	72	68	14	20	68	14	18	3	38	86	56	94	68	14	54	52	34
206	61	95	50	86	68	10	79	52	74	72	14	38	57	10	94	8	14	77	55	72
213	29	53	60	72	6	25	64	29	79	72	72	82	24	60	51	61	72	27	71	72
214	90	68	53	99	8	25	73	68	88	72	67	55	72	68	12	46	88	27	10	72
218	61	54	87	72	63	49	72	78	6	86	29	22	86	15	62	89	6	53	6	72
223	68	68	68	83	8	42	2	57	79	72	92	49	10	60	51	23	72	36	42	72
224	10	4	34	99	6	3	71	74	3	72	79	55	72	68	12	93	68	91	25	79
225	68	68	18	72	29	25	86	52	25	79	56	96	86	14	12	7	3	70	59	79
228	17	54	87	72	63	18	72	75	75	39	29	22	86	51	56	96	75	53	29	72
233	29	68	86	99	8	41	29	29	13	72	92	42	14	76	51	34	72	91	41	72
234	33	34	68	1	8	13	71	74	79	72	79	81	72	11	54	93	3	91	42	79
236	55	55	70	72	65	13	15	75	74	72	74	50	72	99	56	34	88	70	42	72
237	33	88	95	72	33	3	72	71	3	79	63	50	72	95	56	50	95	53	16	72
241	87	4	68	12	94	88	72	87	86	72	93	50	93	93	44	34	46	37	82	57
242	52	89	53	72	94	74	100	52	28	72	46	70	93	74	51	93	72	91	68	72
243	29	89	68	31	42	74	72	50	16	44	72	70	93	22	51	55	72	91	44	72
245	61	34	33	7	42	6	72	73	6	88	3	37	72	27	26	34	3	53	82	19
246	55	61	33	51	94	13	45	75	3	72	79	86	72	93	25	34	95	53	40	72
247	19	83	53	72	77	39	72	99	74	79	10	93	86	94	25	3	56	53	18	72
248	52	33	87	72	77	24	72	69	15	29	10	38	86	54	60	22	52	53	13	72
251	75	42	94	78	63	93	91	52	93	72	46	72	88	93	44	34	46	100	93	57
252	68	85	94	15	40	93	53	87	26	79	66	72	86	93	51	85	72	100	67	70
253	55	85	36	85	16	21	71	26	10	79	46	70	47	60	51	55	73	36	79	73
255	85	34	53	47	42	6	45	6	13	68	72	77	67	43	14	68	95	53	82	15
261	75	17	94	15	63	22	91	88	73	56	88	41	73	93	60	94	46	90	73	20
263	52	34	76	23	93	6	53	87	41	72	95	100	73	75	76	34	50	36	79	92
264	52	34	53	23	67	6	15	76	6	72	79	59	88	84	3	55	79	53	72	77
265	34	34	97	93	34	25	66	75	13	45	73	57	61	75	3	52	73	53	72	16
266	34	34	76	13	34	12	45	75	13	72	79	86	66	94	25	86	95	53	27	40
268	21	61	53	79	93	1	72	92	1	86	50	93	55	44	56	4	79	53	93	4

	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
15	78	37	61	18	9	22	38	66	79	52	79	66	79	15	79	66	97	18	12	38
16	78	72	54	83	60	32	83	5	79	86	79	51	18	27	83	42	7	47	13	35
17	78	18	54	86	25	66	82	44	79	77	83	86	68	31	83	58	97	47	91	58
18	94	18	54	88	55	18	82	3	79	7	79	79	78	1	78	25	80	31	99	83
24	78	12	68	56	8	64	60	33	57	12	79	66	54	29	6	72	6	60	60	38
31	58	43	6	77	76	22	75	66	79	12	73	71	7	49	26	71	9	94	17	81
35	43	54	75	26	91	96	3	44	78	52	83	56	7	27	79	48	36	34	85	38
37	76	56	76	72	36	93	82	82	79	13	51	79	54	74	79	72	68	92	25	37
44	58	53	3	21	6	15	3	73	38	16	99	58	37	27	5	58	39	3	65	61
45	66	54	75	21	91	96	3	44	56	24	36	79	3	29	60	78	46	92	85	38
52	79	72	6	58	7	14	75	33	79	12	78	78	5	2	76	99	6	94	75	45
53	58	52	9	55	3	55	3	79	83	9	99	48	2	99	58	47	94	3	75	61
54	58	53	6	26	3	80	3	73	48	64	99	98	6	30	41	38	91	3	53	61
57	24	54	32	72	75	52	82	61	36	13	56	79	54	62	81	36	23	85	20	42
58	63	78	63	35	23	30	9	74	63	64	36	56	78	9	65	36	23	13	50	31
62	35	72	6	58	13	5	75	48	79	7	48	67	5	7	20	11	8	21	3	5
72	36	72	9	40	3	78	3	48	79	7	58	98	5	7	20	46	8	21	3	73
73	58	68	74	40	3	39	3	33	46	14	37	36	5	12	10	46	10	3	75	47
83	54	30	63	40	53	56	3	33	73	63	38	36	81	13	20	48	46	81	75	4
84	85	53	67	55	3	27	3	73	44	96	48	36	85	84	62	48	35	30	82	4
85	78	85	80	44	3	82	85	73	32	96	56	78	58	52	81	42	48	52	12	73
122	72	41	18	10	15	53	75	34	43	83	86	61	10	83	43	86	78	15	75	5
123	48	53	5	83	78	50	6	33	44	74	58	69	81	64	74	34	23	80	10	32
132	40	57	55	51	5	80	6	33	19	64	34	8	83	71	100	88	31	15	55	5
133	80	53	9	83	45	50	6	33	58	55	39	8	100	64	63	39	83	80	76	81
136	82	72	37	44	5	46	99	100	79	55	44	66	66	100	37	35	37	99	80	100
143	88	6	31	63	10	50	6	34	63	64	20	73	63	64	95	39	95	80	91	64
148	81	63	78	72	15	91	8	25	51	64	99	81	78	75	25	21	42	87	52	23
153	88	11	36	72	10	49	53	20	58	86	56	2	38	45	38	57	38	80	76	91
154	49	68	38	67	10	84	6	33	5	71	33	2	36	86	35	69	35	53	95	6
163	24	53	36	72	76	59	53	100	48	17	2	30	37	45	37	96	4	80	32	88
164	78	68	95	8	47	17	4	100	30	71	33	30	36	47	35	63	4	53	95	11
173	45	52	37	72	74	9	53	20	58	32	14	66	36	7	36	5	30	80	32	52
174	78	57	37	23	48	46	4	87	30	100	33	66	66	48	37	3	58	53	59	59
183	35	52	35	20	74	9	53	20	48	21	75	66	11	21	68	16	80	60	43	45
184	78	57	35	33	7	9	7	87	3	64	67	68	37	71	35	99	55	53	36	52
185	16	57	35	14	71	52	68	100	40	7	14	62	37	23	37	3	37	9	59	52
186	7	72	35	78	75	52	68	80	40	7	44	29	38	23	38	3	19	75	2	31
187	55	63	2	20	74	52	75	70	99	23	74	19	68	81	80	62	81	23	95	81
193	4	52	36	72	74	59	53	100	79	64	48	66	77	9	66	44	39	60	36	52
194	78	57	36	67	73	7	73	87	41	64	67	66	36	22	77	89	97	53	36	46
195	18	57	36	14	75	52	52	12	51	22	48	67	36	65	36	82	2	75	36	47
198	2	72	2	74	48	52	75	37	30	2	95	87	38	75	80	3	19	14	2	4
197	53	63	53	24	73	52	75	82	51	23	34	67	66	95	42	65	19	23	78	24
198	78	63	87	72	73	52	23	3	44	23	41	39	38	81	68	28	39	23	23	29
204	57	57	60	100	74	7	74	48	41	59	67	85	85	71	62	51	8	53	64	17
205	81	57	77	14	75	52	52	85	41	73	14	84	36	76	59	5	4	75	22	44
208	43	72	21	79	84	52	75	62	79	23	95	84	67	9	68	92	18	14	23	77
213	94	52	9	72	88	17	88	26	79	71	79	85	39	71	17	11	28	80	90	11
214	16	57	27	68	74	14	74	51	88	7	68	90	17	4	17	45	63	53	38	11
218	94	83	78	72	55	52	88	3	86	52	79	96	100	17	76	15	86	53	80	76
223	53	52	7	79	76	49	76	44	79	21	79	96	39	21	74	1	8	68	33	11
224	16	52	91	88	75	71	75	28	79	60	79	96	25	4	74	30	8	52	77	77
225	18	57	97	95	75	71	75	3	79	21	100	96	29	59	19	50	19	75	42	55
228	94	57	53	72	94	75	87	26	88	59	79	55	86	67	80	12	86	87	52	27
233	68	52	7	79	75	10	75	44	79	87	79	40	10	82	74	15	19	68	59	33
234	19	52	91	79	75	71	75	14	79	33	79	82	68	4	77	16	19	68	59	85
238	94	72	53	88	25	71	88	74	79	82	15	49	15	83	29	28	68	88	58	55
237	94	72	53	72	75	52	88	14	51	87	18	50	18	48	5	1	86	95	23	76
241	21	70	71	79	52	61	68	39	87	38	92	96	8	92	44	68	39	68	84	75
242	21	72	33	79	74	87	74	48	87	87	72	1	70	95	21	68	12	68	84	36
243	11	87	91	72	75	50	52	45	88	53	72	96	8	4	11	1	8	68	59	97
245	94	57	53	95	68	61	99	26	88	97	14	82	15	76	77	62	16	58	84	97
246	94	72	53	78	73	75	52	51	79	82	49	40	57	24	86	89	15	68	64	97
247	94	72	53	72	73	89	58	75	81	55	42	40	57	47	86	28	88	58	60	59
248	58	49	53	72	33	52	56	68	86	4	72	83	49	30	90	88	11	87	99	27
251	33	57	36	57	44	34	68	46	67	52	92	33	46	3	5	88	18	88	100	61
252	11	52	71	72	67	34	74	72	88	76	92	96	79	92	79	1	8	68	100	97
253	11	68	38	73	24	93	52	45	88	37	73	96	79	80	11	54	8	68	84	97
255	61	87	53	73	75	42	52	83	88	21	48	72	49	2	77	89	8	56	84	59
261	33	57	71	57	60	10	68	46	49	52	8	66	48	3	49	68	18	68	61	75
263	60	87	91	72	75	6	75	7	88	36	79	62	79	73	21	42	24	68	59	59
264	11	57	53	79	56	78	52	58	88	87	79	16	97	22	21	89	40	58	38	59
265	38	57	53	73	52	34	52	46	88	21	41	42	15	5	86	98	15	75	64	59
266	74	57	53	79	10	78	92	74	79	1	72	79	57	63	86	89	75	75	53	34
268	39	57	53	47	35	32	73	11	86	49	47	38	57	27	23	100	93	73	99	63

	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
15	25	18	4	87	20	88	33	52	82	54	55	17	82	43	81	46	93	95	49	37
16	35	4	4	81	16	77	73	73	35	69	83	50	47	23	93	73	93	17	12	61
17	68	62	49	13	33	17	76	4	35	54	17	97	61	23	92	1	100	6	38	61
18	78	62	16	93	17	17	77	76	83	54	38	97	47	23	19	77	22	64	55	43
24	37	60	50	91	17	10	6	24	64	91	88	16	23	36	65	4	93	95	55	68
31	68	17	9	68	78	83	31	76	16	99	95	48	4	36	90	5	88	59	58	94
35	80	85	16	12	64	64	9	52	98	12	82	50	27	24	70	73	88	21	58	38
37	18	82	50	13	17	17	77	77	72	54	94	85	78	43	88	2	23	95	35	23
44	29	60	3	12	91	36	9	75	17	53	31	20	27	73	44	73	95	49	67	3
45	24	85	3	54	91	17	9	52	21	54	42	27	47	72	81	73	95	21	39	3
52	38	75	44	12	78	86	2	2	21	100	90	36	49	36	26	73	90	58	72	94
53	57	60	3	12	32	89	7	44	21	77	80	59	61	58	5	21	90	58	99	3
54	24	60	3	12	32	58	44	75	40	43	90	59	61	73	2	73	90	17	99	3
57	60	14	59	16	17	80	77	77	72	54	61	86	93	88	46	22	91	26	55	31
58	80	75	5	13	88	21	77	77	35	78	61	27	68	63	46	77	46	64	35	33
62	57	3	9	12	74	74	14	14	46	32	90	85	61	36	84	29	70	58	25	94
72	63	3	14	30	86	78	53	14	40	32	62	15	61	36	26	93	70	36	28	94
73	63	60	9	52	89	89	7	50	32	11	8	92	65	58	81	1	95	58	22	14
83	37	3	3	52	37	37	85	85	40	74	5	18	61	1	43	56	43	67	15	23
84	89	3	24	52	38	38	9	9	40	84	5	92	90	1	86	66	11	87	22	14
85	90	85	30	52	98	58	75	8	44	54	82	28	3	47	88	66	85	4	12	76
122	48	75	53	82	59	59	76	45	1	26	44	14	61	78	78	99	85	34	81	11
123	25	75	75	100	36	36	51	64	1	74	14	21	30	23	23	37	84	33	98	11
132	80	15	76	100	83	25	78	45	98	61	2	32	58	30	9	30	85	33	85	23
133	24	8	80	100	16	26	80	53	58	86	77	21	78	84	37	95	91	25	98	30
136	95	16	77	9	97	34	8	75	68	53	94	7	71	23	88	35	97	23	98	95
142	38	6	80	30	28	34	51	80	22	86	29	21	58	95	78	43	91	28	69	30
146	37	8	9	46	81	86	8	87	79	10	70	32	25	75	26	95	4	97	96	48
153	38	53	11	30	25	85	76	17	48	86	27	21	58	83	2	83	82	25	81	35
154	35	53	53	9	19	36	73	9	23	49	27	21	75	37	30	35	31	32	69	2
163	37	53	60	36	16	16	76	17	2	18	82	47	58	38	30	27	53	95	98	9
164	35	4	75	14	95	13	75	17	22	49	65	46	75	43	66	43	53	95	89	22
173	35	53	11	95	33	33	10	45	49	34	29	93	75	87	8	27	76	75	98	22
174	37	4	7	35	95	97	7	7	2	53	29	82	9	3	92	56	53	59	98	22
183	35	53	11	95	3	3	80	21	92	34	29	47	75	98	92	48	4	11	69	27
184	15	7	10	25	81	95	10	9	25	34	50	29	55	76	92	81	60	35	89	85
185	8	75	55	90	72	35	7	7	82	19	50	70	9	60	5	95	81	100	89	48
186	6	23	2	80	86	86	75	75	82	19	82	93	7	83	15	83	53	37	98	27
187	65	75	23	90	79	60	23	23	65	63	1	93	23	75	66	83	90	38	98	92
193	55	68	55	54	3	40	60	7	92	49	37	45	77	32	82	54	36	37	89	22
194	11	68	100	38	81	84	100	9	92	34	82	44	11	31	92	81	11	55	69	23
195	29	75	55	90	86	38	73	73	88	53	91	97	2	60	5	87	4	64	70	44
196	3	75	55	90	86	32	14	14	92	19	65	93	60	2	15	87	81	36	89	91
197	28	75	68	96	32	32	75	23	65	86	65	76	11	23	47	60	36	36	96	65
198	28	23	36	96	45	86	23	23	65	81	85	60	58	23	70	60	54	97	96	60
204	46	68	27	11	67	23	73	14	92	34	77	65	97	95	30	76	11	35	51	91
205	44	75	55	43	86	54	73	73	92	19	91	44	35	54	5	54	4	27	50	91
206	28	75	36	43	83	86	14	14	65	57	77	97	43	22	15	56	53	17	16	35
212	11	68	55	98	100	100	39	75	79	8	27	52	37	95	48	32	33	27	54	91
214	65	68	55	38	76	100	73	7	88	86	77	62	43	94	48	3	33	27	54	27
218	48	88	91	90	38	88	93	93	65	88	77	32	81	95	96	74	98	95	15	48
223	15	68	36	97	72	72	40	33	79	29	70	52	37	78	48	74	27	37	54	36
224	15	68	36	27	100	76	73	11	79	8	36	4	27	1	87	3	59	27	54	97
229	77	75	96	27	83	48	74	21	92	57	36	87	27	5	96	95	70	36	54	36
228	28	87	76	96	85	32	87	87	52	86	77	77	76	95	96	56	64	93	89	27
233	33	68	36	97	100	72	16	58	79	8	81	52	36	50	100	74	59	36	54	91
234	16	75	43	36	55	70	12	21	79	8	55	87	36	90	81	100	97	36	54	91
238	98	68	43	25	43	84	75	70	74	57	91	52	35	83	96	25	64	91	12	91
237	8	88	50	96	90	4	3	95	12	66	91	50	80	58	50	80	91	55	13	91
241	68	68	100	11	37	37	28	97	72	44	93	10	85	10	79	52	97	59	16	100
242	38	68	38	38	81	33	6	97	79	44	36	36	55	35	100	76	36	70	28	91
243	68	68	38	36	81	64	6	85	79	44	7	87	91	4	100	52	84	36	19	91
245	2	68	43	43	81	81	25	21	26	57	36	87	43	76	100	14	55	91	28	91
246	67	68	33	43	35	81	74	14	25	57	43	31	35	17	100	73	84	91	12	91
247	28	58	100	84	53	91	74	95	28	86	90	64	35	17	100	80	64	33	39	91
248	28	58	100	84	53	91	74	35	92	86	80	50	100	17	100	52	83	100	12	83
251	68	68	1	81	81	38	28	61	72	39	90	64	37	78	91	52	97	59	16	100
252	11	68	90	56	38	64	28	61	73	39	92	11	94	78	3	52	97	33	28	91
253	29	68	96	100	37	64	28	11	73	44	93	60	81	41	91	52	97	33	28	91
255	67	68	96	96	43	76	28	59	28	49	37	77	37	17	27	52	84	91	28	91
261	68	68	78	98	81	64	17	61	14	52	86	32	97	65	84	52	97	4	19	91
263	82	75	96	43	54	54	28	81	72	15	77	87	90	66	91	9	97	100	16	91
264	82	52	60	68	35	60	28	61	28	87	92	87	38	78	91	95	59	91	28	91
265	28	52	96	96	43	97	12	59	44	57	43	87	83	17	80	98	83	91	28	91
266	28	86	96	50	70	76	12	95	12	57	91	36	42	17	33	56	61	83	12	91
268	28	73	96	33	53	53	11	73	28	11	64	20	19	11	96	60	83	91	2	84

	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
15	87	70	87	32	12	88	3	60	92	41	6	36	95	29	98	30	62	99	13	29
16	17	98	84	27	98	92	33	11	32	35	73	35	95	13	98	90	96	96	26	27
17	49	89	11	27	89	50	73	11	43	88	62	39	24	27	87	70	96	92	7	27
18	49	98	21	92	98	95	6	86	24	87	62	23	24	95	83	50	98	92	89	27
24	95	27	87	31	60	18	6	53	92	91	95	17	95	27	96	100	95	56	17	29
31	58	87	87	46	59	95	60	62	96	88	59	94	95	49	99	26	83	99	83	85
35	88	27	27	20	34	82	6	3	82	55	5	41	50	59	56	90	28	12	76	27
37	85	27	27	81	86	50	76	29	18	83	62	23	65	17	72	26	87	98	75	27
44	26	27	27	90	89	84	39	12	20	82	81	39	55	27	11	100	30	98	75	14
45	17	29	100	82	89	7	20	24	32	28	94	81	59	59	37	70	87	98	62	27
52	56	87	59	46	99	90	91	71	84	28	3	94	49	65	99	28	62	89	6	50
53	58	86	1	31	99	90	19	53	20	28	81	6	55	89	34	65	92	89	13	30
54	49	24	24	46	85	95	8	12	12	28	81	1	15	99	65	28	90	89	13	30
57	95	29	30	41	98	23	1	96	15	48	59	73	50	17	37	60	87	89	20	21
58	95	29	29	41	29	50	80	89	20	92	59	60	50	49	37	99	99	89	73	21
62	58	28	28	48	99	82	55	99	84	85	49	19	49	35	81	26	17	89	6	27
72	36	81	28	48	70	82	15	77	97	88	82	8	58	99	81	28	60	89	68	82
73	58	28	22	15	99	19	45	6	92	94	82	10	97	99	86	22	3	89	68	32
83	58	35	22	98	99	19	8	3	18	28	19	45	82	99	97	85	94	89	6	81
84	19	100	1	98	99	18	8	3	81	81	82	81	82	48	82	92	92	99	68	21
85	19	17	1	10	98	18	55	54	2	65	82	10	57	49	12	91	85	72	6	44
122	34	92	92	27	59	37	44	86	47	92	3	81	59	48	16	92	88	50	24	100
123	20	27	92	1	36	98	44	36	47	92	3	85	99	28	20	92	85	48	24	100
132	72	1	27	97	59	35	83	59	47	98	3	23	48	26	57	1	88	50	24	33
133	73	94	87	70	97	19	100	42	47	92	3	3	99	28	13	98	82	50	24	33
138	35	89	1	27	31	16	71	3	82	1	89	86	85	25	81	1	92	31	9	8
143	13	94	27	27	34	35	83	52	22	92	35	3	47	85	13	70	27	26	31	33
148	37	94	45	27	21	74	58	81	10	55	90	13	46	87	34	70	92	49	89	100
153	38	77	27	92	88	90	73	86	15	21	90	3	2	85	59	29	92	26	11	100
154	38	92	54	27	48	38	30	52	22	30	81	12	2	28	74	27	92	19	9	33
163	37	29	99	29	52	90	15	60	29	47	90	66	77	34	30	61	82	34	60	100
164	36	87	89	29	60	3	55	52	22	46	66	16	48	71	46	36	92	31	9	100
173	38	35	98	29	9	6	45	78	22	75	90	61	48	34	75	36	29	34	60	100
174	35	81	23	29	71	37	55	57	22	14	66	61	27	71	9	36	29	34	9	33
183	36	94	69	30	9	81	4	9	27	55	81	75	5	28	30	36	65	34	60	100
184	35	94	18	93	21	81	7	4	17	17	94	75	80	71	75	81	29	34	76	33
185	37	36	99	93	4	81	17	66	25	11	94	96	58	22	55	36	65	34	22	100
186	38	35	84	91	31	81	23	66	91	94	8	58	44	23	11	35	82	31	23	100
187	38	53	32	93	87	88	81	83	65	17	58	87	14	23	80	38	92	31	23	100
193	77	94	89	29	7	53	55	4	24	91	38	55	85	34	59	54	65	49	60	100
194	38	38	99	29	7	81	61	4	48	17	38	33	84	7	22	36	82	31	85	25
195	38	90	99	62	16	88	21	68	48	17	94	33	89	73	55	38	65	34	33	100
196	38	90	96	28	93	66	75	68	48	17	3	37	74	14	17	90	92	31	14	100
197	38	90	27	28	65	66	2	65	82	17	58	85	75	95	80	90	92	86	23	100
198	35	66	80	92	91	55	23	34	92	46	2	51	76	14	91	66	92	57	23	98
204	17	54	95	47	2	68	77	4	30	21	100	85	78	97	29	38	70	72	27	33
205	17	54	78	91	23	68	55	68	48	55	14	77	7	73	17	54	45	34	33	25
206	33	54	80	35	80	53	75	17	48	11	14	33	86	14	17	55	65	72	14	25
213	11	54	99	30	71	2	2	29	75	58	56	33	21	38	17	36	51	31	60	33
214	11	54	4	91	7	68	91	68	48	21	22	33	22	48	29	55	88	72	27	25
218	53	90	6	48	58	81	33	48	48	82	7	43	78	53	88	21	3	31	87	90
223	33	49	31	75	21	81	55	66	75	15	22	85	94	48	74	91	85	6	27	100
224	33	84	4	91	33	81	81	66	48	21	14	85	96	33	33	91	78	72	97	33
225	11	98	78	91	34	81	36	76	48	11	88	85	66	100	17	91	4	87	91	100
228	93	84	41	46	53	63	53	91	27	17	75	43	36	87	50	78	71	83	23	98
233	85	100	98	75	52	36	38	66	70	68	61	85	85	33	74	91	85	8	59	100
234	81	100	19	81	21	55	81	68	70	59	22	75	81	70	24	91	88	72	59	100
238	33	70	50	43	84	77	91	81	70	58	25	61	33	58	58	91	88	72	70	22
239	55	53	50	53	53	33	53	78	78	93	75	91	78	93	17	33	85	45	53	50
241	85	4	62	85	87	55	91	75	91	68	100	11	85	50	44	75	85	46	71	100
242	84	25	23	100	68	21	33	67	91	68	22	36	84	59	17	33	85	99	71	100
243	81	81	99	33	52	38	91	88	91	68	67	24	84	97	22	84	85	37	84	100
245	100	53	23	35	52	38	90	33	100	81	73	91	91	17	81	91	2	87	53	100
246	100	53	54	84	81	55	90	83	81	58	58	84	59	58	17	59	85	53	53	100
247	37	90	32	53	95	55	35	83	83	58	75	59	58	58	17	59	92	72	53	100
248	92	53	44	53	83	55	53	100	100	42	88	70	75	93	17	63	60	53	52	100
251	84	84	31	97	88	55	97	37	97	88	48	12	84	64	44	85	85	48	71	100
252	84	91	32	37	81	55	84	68	84	68	95	24	84	84	86	97	87	99	71	100
253	84	4	23	63	52	91	36	66	63	68	67	75	83	37	34	24	52	33	71	37
255	100	53	31	83	34	37	33	97	100	85	73	84	83	21	74	59	77	53	53	100
261	97	59	31	84	94	17	59	94	84	68	3	54	84	4	44	85	85	38	71	100
263	84	71	32	17	38	36	71	35	83	66	73	22	83	92	27	61	52	56	38	100
264	84	91	32	78	97	59	35	88	94	66	73	38	83	21	34	61	52	49	53	91
265	83	4	32	78	78	94	64	55	37	77	73	94	78	21	74	59	36	55	53	91
266	94	100	32	78	78	90	100	100	100	39	88	83	78	22	21	59	87	4	53	53
268	94	78	32	53	81	19	94	53	91	55	54	91	53	29	21	35	92	64	53	83

	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
15	36	93	23	29	96	15	98	9	15	37	92	95	59	55	100	59	32	68	36	96
16	41	92	84	27	25	27	98	93	27	92	93	95	13	55	100	63	64	68	36	96
17	39	31	11	65	25	27	96	99	27	47	92	17	99	55	100	83	26	68	35	96
18	29	24	84	50	29	92	99	88	28	28	26	99	88	60	100	62	18	68	35	78
24	39	26	23	27	49	27	96	50	27	52	95	20	50	49	28	59	60	68	36	55
31	83	46	76	2	49	49	95	84	27	14	50	26	9	88	88	35	83	76	79	5
35	23	26	39	27	24	27	24	75	27	12	49	49	29	56	42	17	52	75	42	79
37	39	23	53	28	47	27	17	97	74	35	67	17	97	55	23	17	85	43	42	96
44	31	70	51	52	58	27	79	13	24	53	95	86	27	43	42	73	75	75	69	58
45	62	90	53	12	76	29	79	52	87	12	95	49	29	58	42	38	52	75	35	46
52	50	95	11	85	88	4	98	52	16	28	50	28	100	26	90	96	13	68	36	73
53	62	95	84	52	89	86	36	91	47	26	93	49	82	28	90	49	60	68	58	55
54	59	95	84	52	89	100	79	75	52	52	88	40	24	43	42	49	75	75	73	96
57	78	23	24	16	35	70	17	29	60	38	52	17	59	52	91	17	28	68	58	83
58	26	23	46	64	38	27	32	46	20	89	14	52	21	98	64	73	64	41	35	78
62	91	49	26	85	40	28	38	52	65	65	64	49	96	88	95	96	17	68	35	73
72	16	62	77	85	46	65	36	52	65	65	60	27	84	68	95	47	29	8	35	73
73	20	90	77	85	17	28	89	9	65	81	90	33	29	26	95	33	81	68	73	74
83	97	95	82	85	81	11	78	82	65	57	90	57	94	88	70	33	61	68	58	44
84	19	95	98	52	81	57	58	82	92	57	68	87	30	68	70	47	10	8	29	21
85	19	85	96	53	12	83	78	30	22	12	91	87	83	85	91	51	82	6	46	56
122	10	98	14	53	86	25	89	94	92	92	78	1	9	67	81	25	28	54	2	5
123	81	65	1	3	16	22	89	94	27	92	84	1	24	27	94	52	11	97	61	51
132	10	96	29	53	34	25	87	82	97	28	31	29	80	4	65	5	37	54	87	51
133	47	92	14	50	16	89	89	65	70	98	95	14	24	70	1	1	3	54	93	51
136	94	1	36	95	61	89	96	65	54	1	95	56	95	68	61	22	4	3	30	66
143	22	70	48	50	12	35	28	65	97	29	95	29	52	62	4	40	60	57	84	51
146	3	70	24	91	34	27	48	94	54	92	56	17	17	21	34	87	17	74	35	58
153	14	65	85	50	34	97	69	65	27	27	95	27	91	61	4	85	62	57	82	44
154	14	92	90	53	34	90	94	65	97	27	95	2	31	6	81	97	76	98	76	33
163	45	29	83	24	46	77	89	52	27	77	37	27	80	8	1	90	31	83	48	44
164	73	29	3	17	88	48	89	45	27	27	95	27	80	63	69	85	4	96	11	73
173	45	5	88	24	84	84	88	76	95	1	100	27	7	18	98	97	95	56	80	74
174	58	48	77	61	77	77	96	34	32	23	95	48	54	4	69	84	4	56	49	33
183	70	44	83	73	66	87	69	84	36	22	27	27	52	94	54	84	37	98	45	74
184	77	15	35	7	58	87	90	84	35	23	22	2	54	4	51	95	11	56	47	67
185	94	29	97	22	97	90	15	53	97	60	46	76	99	68	51	90	11	41	53	13
186	88	3	4	44	58	90	96	19	36	15	3	7	59	68	45	95	80	56	29	75
187	67	1	84	46	67	53	79	34	54	71	44	46	59	61	41	14	17	71	97	86
193	81	44	37	73	85	54	50	4	38	80	97	82	52	81	87	38	11	56	4	48
194	59	44	37	73	90	36	89	84	37	99	44	97	59	36	51	38	91	56	53	67
195	1	29	43	75	68	36	5	53	90	60	93	2	95	68	51	38	11	41	53	14
196	67	82	83	44	37	54	98	19	90	80	44	91	54	68	41	95	97	41	29	74
197	85	92	83	2	6	54	98	34	90	60	64	91	84	61	50	95	70	58	86	34
198	54	6	5	46	25	88	18	34	90	71	45	14	84	81	41	43	28	56	4	41
204	93	46	38	28	15	54	51	84	38	99	27	27	59	4	16	54	43	56	53	73
205	1	44	45	17	58	54	50	4	27	60	27	48	95	68	65	38	11	10	4	88
208	8	26	83	17	86	54	89	29	84	80	77	48	80	61	41	95	77	56	57	74
213	51	47	82	74	93	54	50	85	9	3	27	98	52	49	98	71	38	48	34	25
214	42	35	44	74	56	54	5	88	70	76	27	70	76	94	96	7	38	92	53	68
218	3	18	57	58	24	54	15	27	84	21	47	53	80	61	3	53	46	47	77	79
223	51	37	8	74	45	54	50	2	32	72	27	4	87	50	5	95	38	79	55	39
224	69	45	8	4	99	54	1	2	23	93	38	7	4	40	98	90	38	88	11	25
225	35	47	45	34	68	54	54	2	32	96	91	48	18	40	82	100	84	88	34	56
228	3	79	57	87	24	54	6	91	84	70	48	87	31	5	28	23	77	24	77	10
233	15	81	8	74	45	81	54	2	43	100	91	70	52	18	87	4	91	79	11	10
234	93	47	19	74	44	5	54	2	43	81	91	70	4	54	22	70	91	68	11	88
238	62	80	4	3	8	50	12	53	84	91	91	95	84	12	85	95	91	8	34	25
237	85	29	51	3	29	22	18	81	84	53	91	95	50	50	50	23	91	30	81	51
241	1	37	12	74	23	68	17	98	1	100	100	100	87	10	89	100	100	83	11	26
242	1	33	17	21	45	68	28	84	84	84	33	10	52	38	3	3	91	79	75	26
243	89	91	77	88	23	87	25	90	38	84	84	84	87	38	4	84	91	66	11	28
245	82	27	18	74	86	89	28	84	84	81	91	37	4	50	42	100	91	23	55	10
246	54	47	15	88	47	43	12	53	90	76	91	95	84	50	83	100	91	30	27	13
247	89	30	31	88	30	46	39	91	90	53	91	35	84	20	3	53	91	30	83	51
248	3	79	49	21	99	93	39	83	84	53	83	38	4	83	89	53	83	13	83	12
251	93	97	18	78	23	87	28	90	59	59	59	52	84	92	84	100	32	34	26	
252	42	38	17	60	23	68	28	27	37	84	33	100	87	84	83	84	100	98	34	6
253	89	27	8	88	86	60	28	100	90	84	84	76	52	90	17	84	84	96	34	6
255	1	24	8	88	86	32	26	81	97	83	91	3	33	46	17	100	91	67	17	41
261	47	59	17	52	45	37	32	100	84	84	33	84	43	84	54	90	84	2	94	14
263	19	15	17	21	47	60	54	15	84	84	100	84	37	100	42	84	84	96	17	14
264	94	45	77	52	67	10	26	81	80	83	83	76	34	76	100	76	91	67	84	73
265	26	29	54	66	35	9	12	81	83	76	97	84	91	76	77	76	83	2	55	41
266	26	15	18	21	4	9	12	53	84	76	83	94	91	61	77	100	83	24	19	44
268	41	1	19	12	47	9	24	83	84	53	81	37	59	61	74	53	83	1	59	3

Table D1: Full Classification Matrix